Red Hat Satellite 6.0
User Guide

A guide to using Satellite entitlement management software.

Red Hat Satellite Documentation Team

Red Hat Satellite 6.0 User Guide

A guide to using Satellite entitlement management software.

Red Hat Satellite Documentation Team

Legal Notice

Copyright © 2014 Red Hat.

This document is licensed by Red Hat under the Creative Commons Attribution-ShareAlike 3.0 Unported License. If you distribute this document, or a modified version of it, you must provide attribution to Red Hat, Inc. and provide a link to the original. If the document is modified, all Red Hat trademarks must be removed.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, JBoss, MetaMatrix, Fedora, the Infinity Logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux ® is the registered trademark of Linus Torvalds in the United States and other countries.

Java ® is a registered trademark of Oracle and/or its affiliates.

XFS ® is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL ® is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js ® is an official trademark of Joyent. Red Hat Software Collections is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack ® Word Mark and OpenStack Logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

Abstract

The Red Hat Satellite 6.0 User Guide describes how to use Satellite, including subscriptions, content management, provisioning, and system control.

Table of Contents

Chapter 1. Introduction to Red Hat Satellite **5**
1.1. Red Hat Satellite 6 System Architecture 5
1.2. Red Hat Satellite 6 Layout and Topology 6
1.3. Red Hat Satellite Server 6 Basic Configuration Workflow 10

Chapter 2. Accessing Red Hat Satellite **13**
2.1. Logging in to Red Hat Satellite 13
2.2. Changing the Password in Red Hat Satellite 14

Chapter 3. Configuring Organizations, Locations and Lifecycle Environments **16**
3.1. Organizations 16
3.1.1. Creating an Organization 17
3.1.2. Creating an Organization Debug Certificate 17
3.1.3. Editing an Organization 18
3.1.4. Removing an Organization 18
3.2. Locations 19
3.2.1. Creating a Location 19
3.2.2. Editing a Location 19
3.2.3. Removing a Location 20
3.3. Lifecycle Environments 20
3.3.1. Creating a Lifecycle Environment 21
3.3.2. Removing a Lifecycle Environment 21

Chapter 4. Using Content Management **23**
4.1. Using the Red Hat Satellite Content Dashboard 23
4.2. Connected Satellite 24
4.2.1. Using Red Hat Content Providers 24
4.2.1.1. Setting up a Manifest 24
4.2.1.2. Uploading a Subscription Manifest 25
4.2.1.3. Enabling Red Hat Repositories 26
4.2.2. Using Products 26
4.2.2.1. Creating a Product 26
4.2.2.2. Adding Repositories to a Product 27
4.2.2.3. Using Bulk Actions for Products 28
4.2.2.4. Using Repository Discovery 28
4.2.2.5. Removing a Product 29
4.2.3. Synchronizing Content 29
4.2.3.1. Synchronization Status 29
4.2.3.2. Creating a Synchronization Plan 30
4.2.3.3. Applying a Synchronization Schedule 30
4.3. Disconnected Satellite 31
4.3.1. Configuring the Synchronization Host 32
4.3.2. Synchronizing Content 34
4.3.3. Exporting Content 36
4.3.4. Importing Content to a Disconnected Satellite Server 36

Chapter 5. Using Content Views **39**
5.1. Creating a Content View 39
5.2. Adding Repositories to the Content View 40
5.3. Filtering Content 40
5.3.1. Creating a Filter 40
5.3.2. Adding Content to a Filter 41
5.3.3. Removing Content from a Filter 42

5.3.3. Removing Content from a Filter 42
5.3.4. Removing a Filter 43
5.4. Publishing a Content View 43

Chapter 6. Searching for Content 45
6.1. Basic Content Search 45
6.2. Content Comparison across Environments 45

Chapter 7. Configuring Activation Keys 46
7.1. Creating an Activation Key 46
7.2. Removing an Activation Key 46
7.3. Adding Objects to Activation Keys 47
7.3.1. Adding Subscriptions to an Activation Key 47
7.3.2. Adding Host Collections to an Activation Key 47
7.4. Removing Objects from Activation Keys 47
7.4.1. Removing Subscriptions from an Activation Key 48
7.4.2. Removing Host Collections from an Activation Key 48

Chapter 8. Configuring GPG Keys 49
8.1. Creating a GPG Key 49
8.2. Removing a GPG Key 49

Chapter 9. Configuring the Provisioning Environment 50
9.1. Host Groups 50
9.1.1. Creating a Host Group 50
9.2. Parameters 51
9.2.1. Creating a Global Simple Parameter 52
9.2.2. Creating a Puppet Class 52
9.2.3. Configuring Smart Parameters 52
9.2.4. Importing Parameterized Classes from a Puppet Master 53
9.2.5. Configuring Parameterized Classes 53
9.3. Configuring Provisioning Settings 54
9.3.1. Domains 54
9.3.1.1. Creating a Domain 54
9.3.2. Subnets 55
9.3.2.1. Creating a Subnet 55
9.3.3. Architectures 56
9.3.3.1. Creating an Architecture 56
9.3.4. Compute Resources 56
9.3.4.1. Creating a Compute Resource 57
9.3.5. Hardware Models 58
9.3.5.1. Creating a Hardware Model 58
9.3.6. Installation Media 59
9.3.6.1. Adding Installation Media 59
9.3.7. Operating Systems 59
9.3.7.1. Adding an Operating System 59
9.3.8. Partition Tables 60
9.3.8.1. Defining a New Partition Table 60
9.3.9. Provisioning Templates 60
9.3.9.1. Creating a Provisioning Template 61

Chapter 10. Configuring Hosts 62
10.1. Creating a Host 62
10.2. Configuring Hosts for Registration 62
10.3. Configuration Options 63

10.3.1. Automated Configuration 63
10.3.2. Manual Configuration 64
10.4. Registration 64
10.4.1. Registering a Host 64
10.4.2. Installing the Katello Agent 66
10.4.3. Installing and Configuring the Puppet Agent 66
10.5. Removing a Host 68

Chapter 11. Using the Foreman Discovery Plugin 69
11.1. Installing the Foreman Discovery Plugin 69
11.2. Configuring the Foreman Discovery Plugin 69
11.3. Editing the Discovered Host 71
11.4. Troubleshooting for the Foreman Discovery Plugin 71

Chapter 12. Configuring Host Collections 73
12.1. Creating a Host Collection 73
12.2. Adding Hosts to a Host Collection 73
12.3. Adding Content to Host Collections 74
12.3.1. Adding Packages to a Host Collection 74
12.3.2. Adding Errata to a Host Collection 74
12.4. Removing Content from a Host Collection 75
12.5. Changing the Assigned Lifecycle Environment or Content View for a Host Collection 75
12.6. Removing a Host from a Host Collection 76
12.7. Removing a Host Collection 76
12.8. Cloning a Host Collection 77
12.9. Reviewing Host Collection Actions 77

Chapter 13. Red Hat Satellite Capsule Servers 78
13.1. Red Hat Satellite Capsule Server Scalability 78
13.2. Red Hat Satellite Capsule Server Prerequisites 79
13.3. Installing a Red Hat Satellite Capsule Server 82
13.4. Configuring a Red Hat Satellite Capsule Server 83
13.5. Adding Lifecycle Environments to a Red Hat Satellite Capsule Server 87
13.6. Removing Lifecycle Environments from the Red Hat Satellite Capsule Server 88
13.7. Registering Host Systems to a Red Hat Satellite Capsule Server 89
13.8. Refreshing a Red Hat Satellite Capsule Server 90

Chapter 14. Users and Roles 91
14.1. Configuring LDAP Authentication for Red Hat Satellite 91
14.2. Users 92
14.2.1. Creating a User 92
14.2.2. Editing a User 92
14.2.3. Removing a User 93
14.3. Roles 93
14.3.1. Creating a Role 93
14.3.2. Adding Permissions to an Existing Role 94
14.3.3. Removing a Role 94

Chapter 15. Command Line Reference 95
15.1. Using the Red Hat Satellite Command-Line Interface 95

Chapter 16. Maintenance 96
16.1. Logging and Reporting 96
16.1.1. Viewing Import History 96
16.2. Disaster Recovery 97

16.2.1. Red Hat Satellite Backup Procedure 97
16.2.2. Red Hat Satellite Restore Procedure 100
16.3. Troubleshooting 102
16.3.1. Changing Your Red Hat Satellite's Fully Qualified Domain Name (FQDN) 102

Chapter 17. Configuring Identity Management in Red Hat Satellite 104
17.1. Configuring Red Hat Satellite Server or Capsule Server for IDM Realm Support 104
17.2. Adding Hosts to an IDM Host Group 106

Chapter 18. Red Hat Satellite User Interface Plugins 108
18.1. Red Hat Access Plugin 108
18.1.1. Searching for Solutions in the Red Hat Access Plugin 108
18.1.2. Utilizing Logs in the Red Hat Access Plugin 108
18.1.3. Viewing Existing Support Cases Using the Red Hat Access Plugin 109
18.1.4. Modifying Existing Support Cases Using the Red Hat Access Plugin 110
18.1.5. Creating New Support Cases Using the Red Hat Access Plugin 110

Glossary of Terms 112

Revision History 116

Chapter 1. Introduction to Red Hat Satellite

Red Hat Satellite 6 is the evolution of Red Hat's life cycle management platform. It provides the capabilities that administrators have come to expect in a tool focused on managing systems and content for a global enterprise. Satellite 6 covers the use cases requested by Satellite 5 customers, but also includes functionality that enables larger scale, federation of content, better control of systems during the provisioning process, and a much more simplified approach to life cycle management. Satellite 6 also further evolves the inherent approach to certificate-based entitlements and integrated subscription management. Satellite 6 is based on years of customer feedback and is an evolution of previous versions.

Report a bug

1.1. Red Hat Satellite 6 System Architecture

Red Hat Satellite 6 is based upon several open source projects arranged in the following architecture.

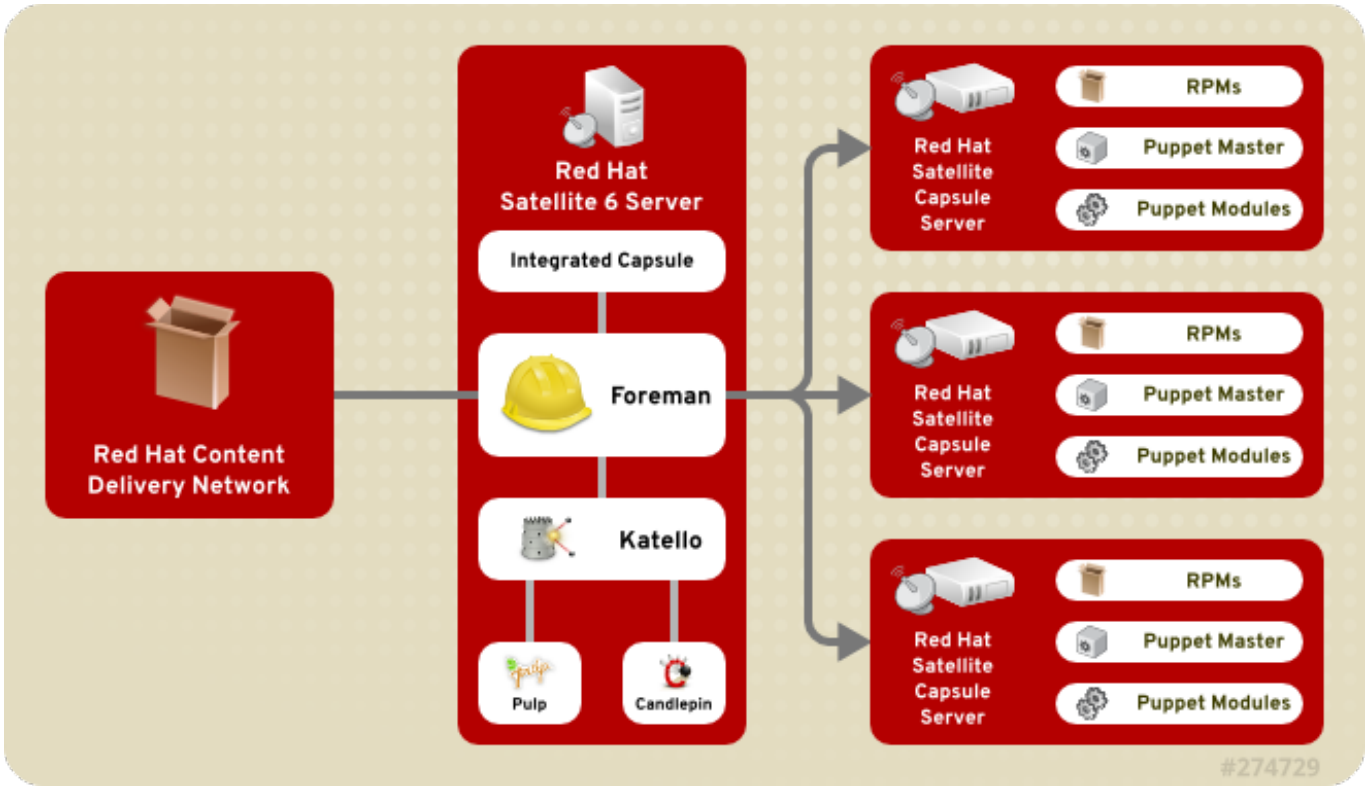


Figure 1.1. Red Hat Satellite 6 System Architecture

Foreman

Foreman is an open source application used for provisioning and life cycle management of physical and virtual systems. Foreman automatically configures these systems using various methods, including kickstart and Puppet modules. Foreman also provides historical data for reporting, auditing, and troubleshooting.

Katello

Katello is a subscription and repository management application. It provides a means to subscribe to Red Hat repositories and download content. You can create and manage different versions of this content and apply them to specific systems within user-defined

different versions of this content and apply them to specific systems within user-defined stages of the application life cycle.

Candlepin

Candlepin is a service within Katello that handles subscription management.

Pulp

Pulp is a service within Katello that handles repository and content management.

Hammer

Hammer is a CLI tool that provides command line and shell equivalents of most Web UI functions.

REST API

Red Hat Satellite 6 includes a RESTful API service that allows system administrators and developers to write custom scripts and third-party applications that interface with Red Hat Satellite.

Capsule

Red Hat Satellite Capsule Server acts as a proxy for some of the main Satellite functions including repository storage, **DNS**, **DHCP**, and Puppet Master configuration. Each Satellite Server also contains integrated Capsule Server services.

Report a bug

1.2. Red Hat Satellite 6 Layout and Topology

The Red Hat Satellite infrastructure layout has to be considered prior to installing Red Hat Satellite 6. Determining the organization of your infrastructure helps align the Satellite Server and Satellite Capsule Servers to best serve your requirements. The following topology diagrams provide examples of infrastructure layout.

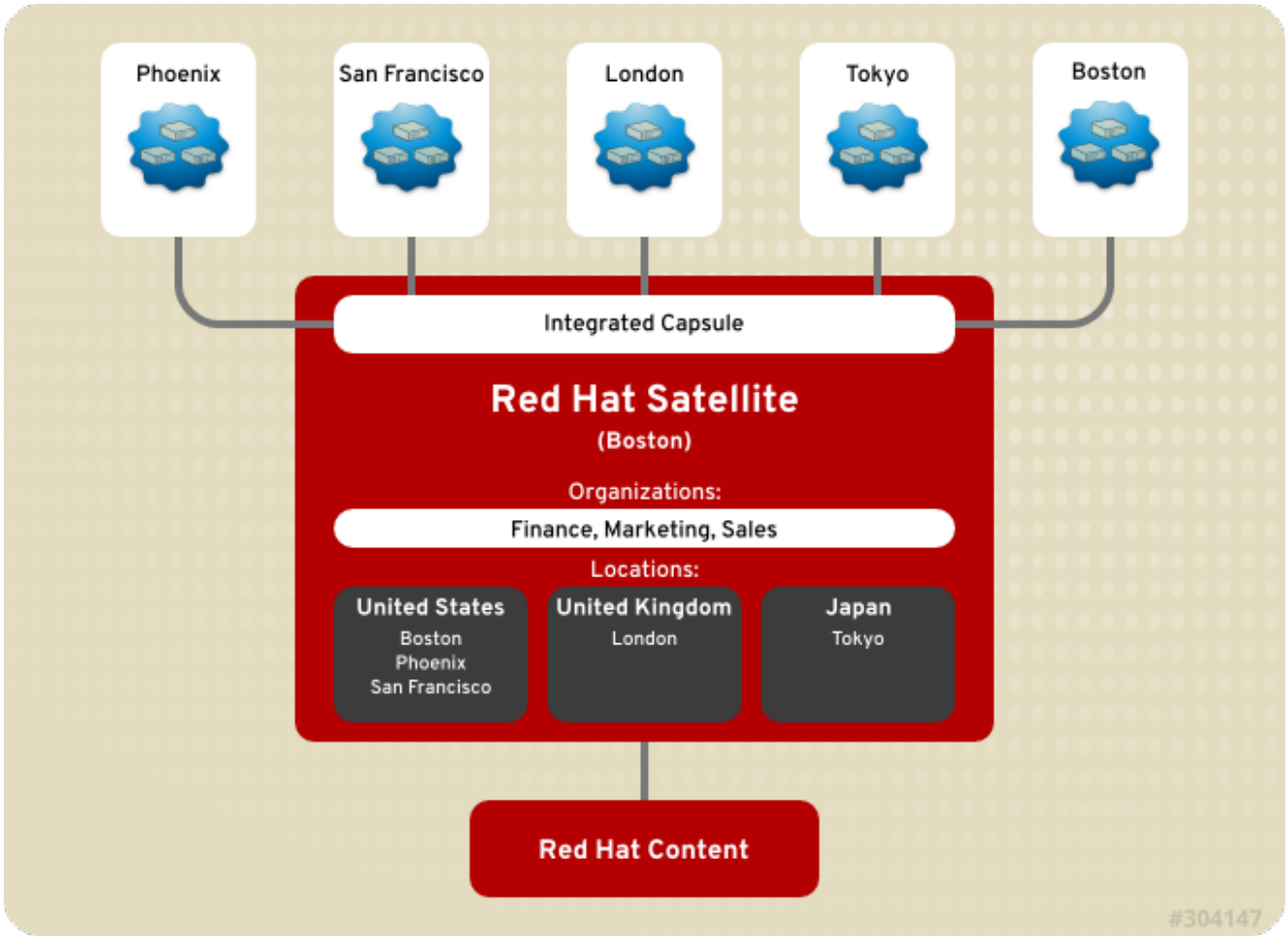


Figure 1.2. Single Satellite with Integrated Capsule

This topology demonstrates the basic use of Red Hat Satellite 6. In this example, there are five pools of hosts registered to the Satellite Server. Red Hat Satellite 6 categorizes these pools into three locations: United States, United Kingdom, and Japan. In addition, each department uses a distinct organization: Finance, Marketing, and Sales. All Satellite Server functions are shared among these Locations and Organizations.

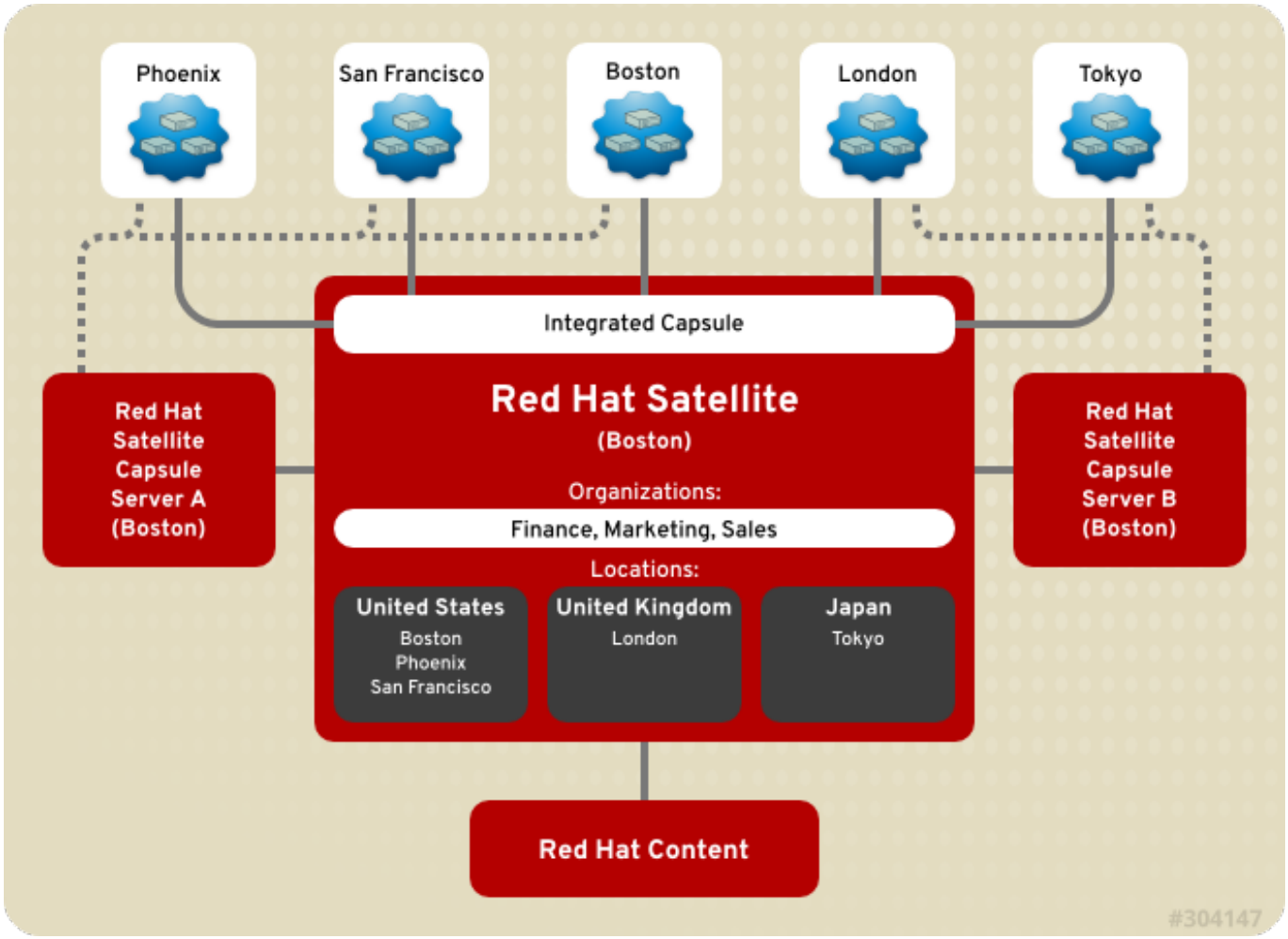


Figure 1.3. Single Satellite with Integrated Capsule and Backup Capsules

This topology shows the addition of two backup Satellite Capsule Servers based in Boston. One Capsule Server caters to the three US-based offices: Phoenix, San Francisco, and Boston. The other caters to the international offices: London and Tokyo. Assigning one backup Capsule Server to the United States location and the other to the United Kingdom and Japan locations, the load reduces on the main Satellite Server and its integrated Capsule. As new offices are created in these locations, the Satellite Server can add them to their location categories and the Capsule Server services the new offices.

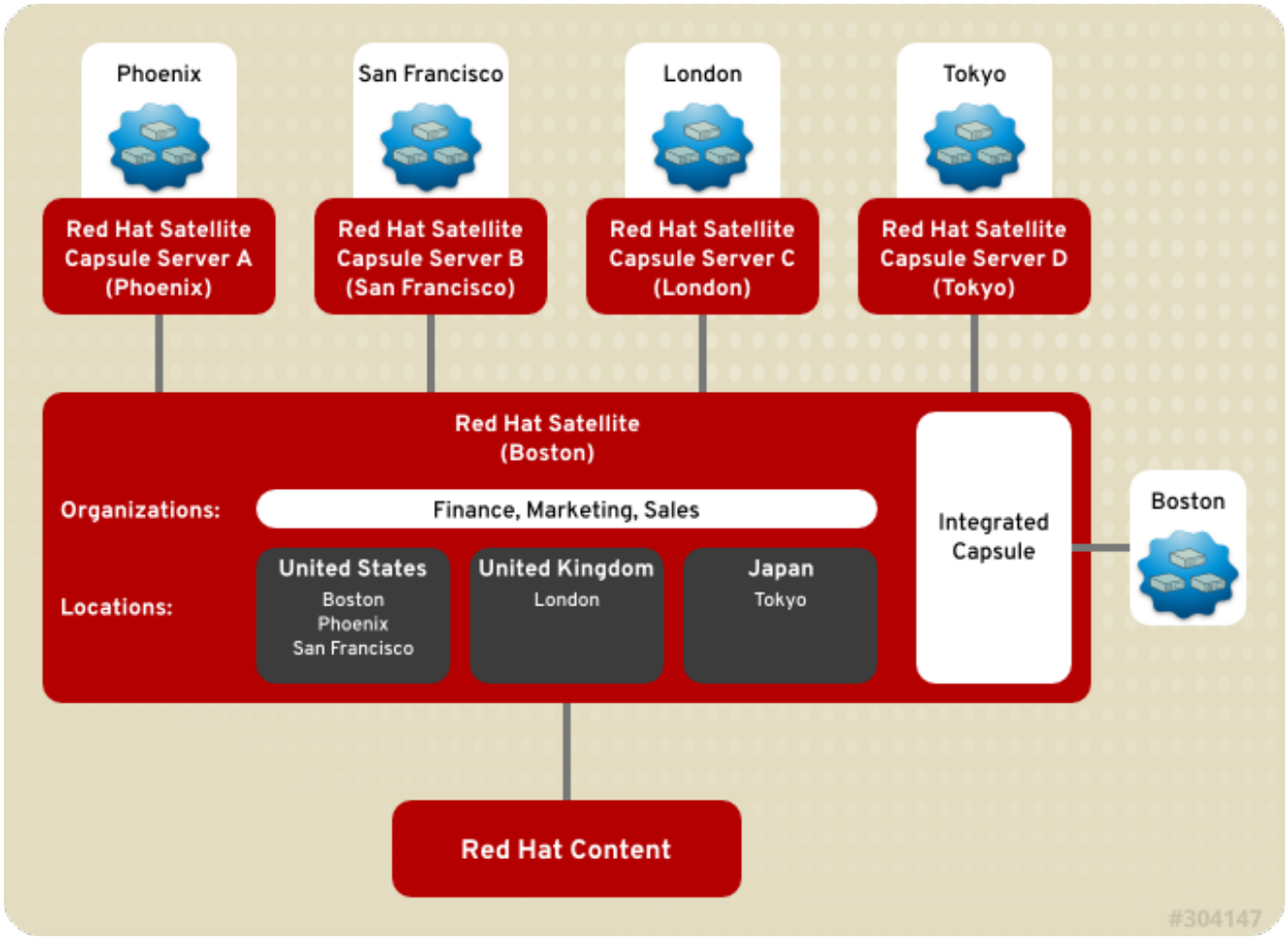


Figure 1.4. Remote Capsules Based on Location

This topology assigns Satellite Capsule Servers to specific locations. The Satellite Server can create a hierarchy of locations; for example, having cities attached to a country. The Satellite Capsule Servers based in these locations are registered to the central Satellite Server in Boston and assigned to their respective locations. Each Capsule services all hosts in each respective location.

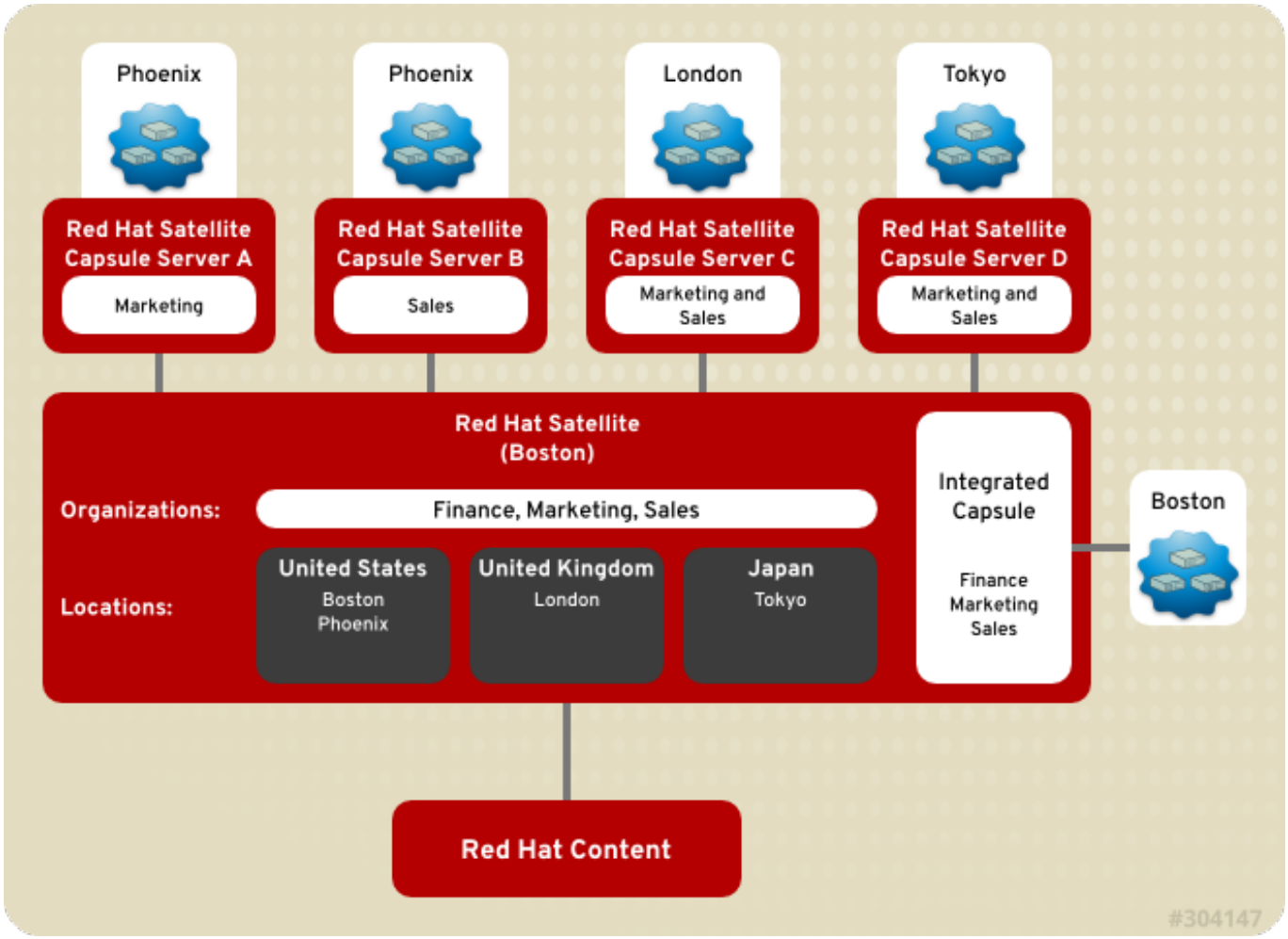


Figure 1.5. Remote Capsules Based on Location and Organization

This topology demonstrates Satellite Capsule Servers assignment to organizations. For example, two Capsule servers are assigned to the Phoenix location, but to different organizations: one for Marketing and the other for Sales. Likewise, two Satellite Capsule Servers are assigned to two organizations: one for both the Sales and Marketing organizations in London, and the other for both the Sales and Marketing organizations in Tokyo. This shows how the combination of Satellite Server and associated Satellite Capsule Servers can manage the layout of multiple organizations in multiple locations working together.

Report a bug

1.3. Red Hat Satellite Server 6 Basic Configuration Workflow

Prerequisites

Before continuing with this workflow you must have successfully installed a Red Hat Satellite 6 Server and any additional required remote capsules. See the *Red Hat Satellite 6 Server Installation Workflow* in the *Red Hat Satellite 6 Installation Guide* for further information.

Initial Configuration

These are the initial procedures to configure a basic Red Hat Satellite Server:

1. Log in to the Satellite Server. This requires the administration user and password. See Section 2.1, “Logging in to Red Hat Satellite” for more information.

a. For more information about changing the password see Section 2.2, “Changing the Password in Red Hat Satellite”.

2. Edit the Red Hat Satellite Integrated Capsule Server to select the desired organizations and locations. The name of the Satellite Integrated Capsule Server will be the same as the hostname of the server that Satellite 6 Server is installed on. See Section 3.1.3, “Editing an Organization” and Section 3.2.2, “Editing a Location” for more information.

3. Edit the desired location to select the resources to be associated with that location. See Section 3.1, “Organizations” for more information.

4. Edit the default organization to select the resources to be associated with that organization. See Section 3.2, “Locations” for more information.

5. Refresh the Satellite Capsule Server. See Section 13.8, “Refreshing a Red Hat Satellite Capsule Server” for more information.

Configuring a Red Hat Satellite Server

These are the initial procedures to configure a basic Red Hat Satellite Server:

1. Create a domain. See Section 9.3.1, “Domains” for more information.

2. Create a subnet. See Section 9.3.2, “Subnets” for more information.

3. Create the desired lifecycle environments. See Section 3.3, “Lifecycle Environments” for more information.

4. Create any desired custom products. See Section 4.2.2.1, “Creating a Product” for more information.

5. Choose the desired Red Hat Repositories.

    a. Create a manifest from the Red Hat Customer Portal. See Section 4.2.1.1, “Setting up a Manifest” for more information.

    b. Upload the manifest in the Satellite Server web interface. This will propagate the subscription information into the Satellite Server. See Section 4.2.1.2, “Uploading a Subscription Manifest” for more information.

    c. Once the manifest has been uploaded, the Red Hat Repositories available from valid Red Hat Subscriptions are imported into the Satellite Server. Choose which repositories are relevant to your organization. See Section 4.2.1.2, “Uploading a Subscription Manifest” for more information.

    d. Optional:

        a. Red Hat source repositories update content based on security errata, bug fixes, and enhancements. To ensure that the Satellite Server is updated automatically, Section 4.2.3.2, “Creating a Synchronization Plan” and Section 4.2.3.3, “Applying a Synchronization Schedule” are recommended practices.

6. Manually synchronize content. See Section 4.2.3.1, “Synchronization Status” for more information.

7. Create a content view with the desired repositories, puppet modules, and filters. Publish the content view then promote it to other lifecycle environments as required. See Chapter 5, *Using Content Views* for more information.

8. Optional:

    a. Create a host collection and assign it to the desired lifecycle environment and content view. See Chapter 12, *Configuring Host Collections* for more information.

9. Create an activation key assigning it to the desired lifecycle environment and content view. See Section 7.1, “Creating an Activation Key” for more information.

10. Edit an existing provisioning template and associate it with the previously created operating system. See Section 9.3.9, “Provisioning Templates” for more information.

11. Edit the operating system created by default when creating the content view with the desired details and ensure it is associated with the desired partition table and provisioning template. See Section 9.3.7, “Operating Systems” for more information.

12. Create a installation medium with the desired details. Ensure that the media is associated with the required locations and organizations. See Section 9.3.6, “Installation Media” for more information.

13. Create a host group with the desired details. See Section 9.1.1, “Creating a Host Group” for more information.

Creating a Backup of a Red Hat Satellite Server

This is the procedure to create a backup of the Red Hat Satellite Server:

1. Create a backup of the Satellite Server containing the required configuration files, data files, repositories, and databases. See the Section 16.2.1, “Red Hat Satellite Backup Procedure” for more information.

Report a bug

Chapter 2. Accessing Red Hat Satellite

2.1. Logging in to Red Hat Satellite

After **Red Hat Satellite** has been installed and configured use the web user interface to log in to **Satellite** for further configuration.

These steps show how to log in to Red Hat Satellite.

1. Access the **Satellite** server using a web browser pointed to the following address:

   `https://HOSTNAME/`

   To identify your hostname, use the **`hostname`** command at the prompt:

   ```
   # hostname
   ```

   Important

   An untrusted connection warning appears on your web browser when accessing **Satellite** for the first time. Accept the self-signed certificate and add the **Satellite** URL as a security exception to override the settings. This procedure might differ depending on the browser being used.

   Only do this if you are sure that the **Satellite** URL is a trusted source.

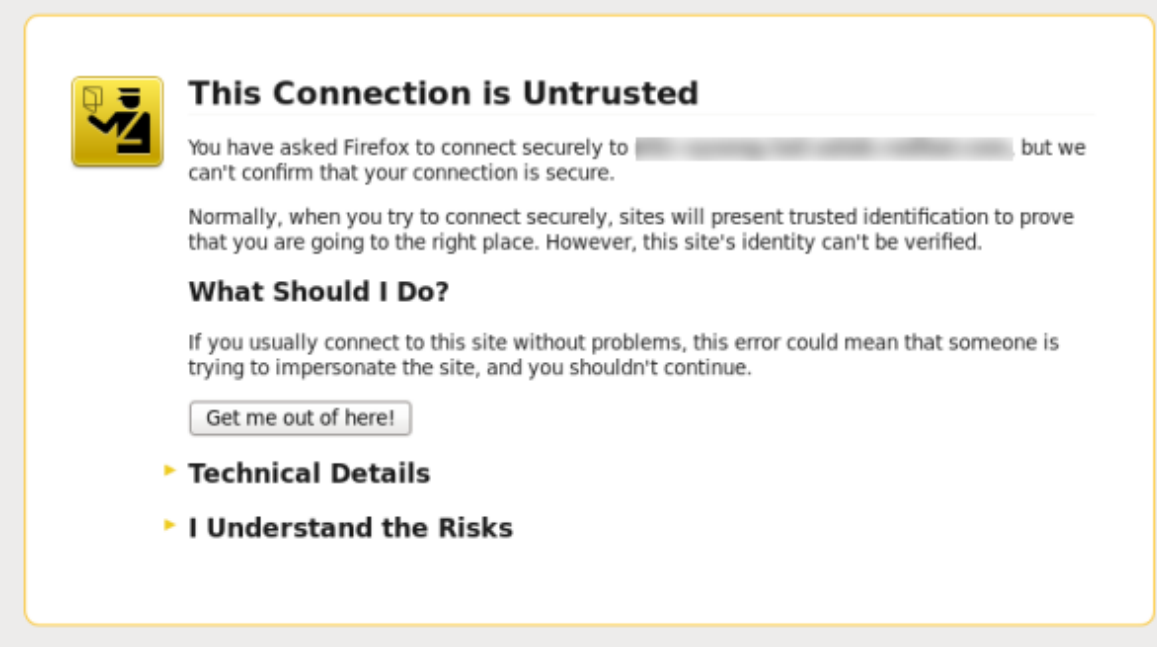
   
   This Connection is Untrusted
You have asked Firefox to connect securely to [redacted] but we can't confirm that your connection is secure.
Normally, when you try to connect securely, sites will present trusted identification to prove that you are going to the right place. However, this site's identity can't be verified.
What Should I Do?
If you usually connect to this site without problems, this error could mean that someone is trying to impersonate the site, and you shouldn't continue.
Get me out of here!
Technical Details
I Understand the Risks

   Figure 2.1. Untrusted Connection Warning

2. Enter the user name and password created during the configuration process. If a user was not created during the configuration process, the default user name is *admin*.

Result

When you have successfully logged in, you are taken to the **Satellite** dashboard. The dashboard contains an overview of the **Satellite** and the hosts registered.

The main navigation tabs are as follows:

Table 2.1. Navigation Tabs

Navigation Tabs	Description
Organization@Location	Clicking this tab changes the organization and location. If no organization or location is selected, the default organization is *Any Organization* and the default location is *Any Location*. Use this tab to change to different values.
Monitor	Provides summary dashboards and reports.
Content	Provides content management tools. This includes Content Views, Activation Keys, and Lifecycle Environments.
Hosts	Provides host inventory and provisioning configuration tools.
Configure	Provides general configuration tools and data including Host Groups and Puppet data.
Infrastructure	Provides tools on configuring how Satellite 6 interacts with the environment.
Administer	Provides advanced configuration for settings such as Users and RBAC, as well as general settings.
User Name	Provides user administration where users can edit their personal information.

Note

If you have forgotten the administrative password, log on to the **Satellite** command-line interface to reset the administration user and password:

```
# foreman-rake permissions:reset
Reset to user: admin, password: qwJxBptxb7Gfcjj5
```

This will reset the password of the default user *admin* to the one printed on the command line. Change this password upon logging in to prevent any security issues from occurring.

Report a bug

2.2. Changing the Password in Red Hat Satellite

These steps show how to change your password.

Procedure 2.1. Changing Password

1. Click your user name at the top right corner.
2. Select **My Account** from the menu.
3. Type in a new password in the **Password** field.

4. Type in the new password again in the **Verify** field.
5. Click the **Submit** button to save your new password.

Result:

The password is changed.

Report a bug

Chapter 3. Configuring Organizations, Locations and Lifecycle Environments

Red Hat Satellite 6 takes a consolidated approach to Organization and Location management. System administrators define multiple Organizations and multiple Locations in a single Satellite server. For example, a company might have three Organizations (Finance, Marketing, and Sales) across three countries (United States, United Kingdom, and Japan). In this example, the Satellite server manages all Organizations across all geographical Locations, creating nine distinct contexts for managing systems. In addition, users can define specific locations and nest them to create a hierarchy. For example, Satellite administrators might divide the United States into specific cities, such as Boston, Phoenix, or San Francisco.

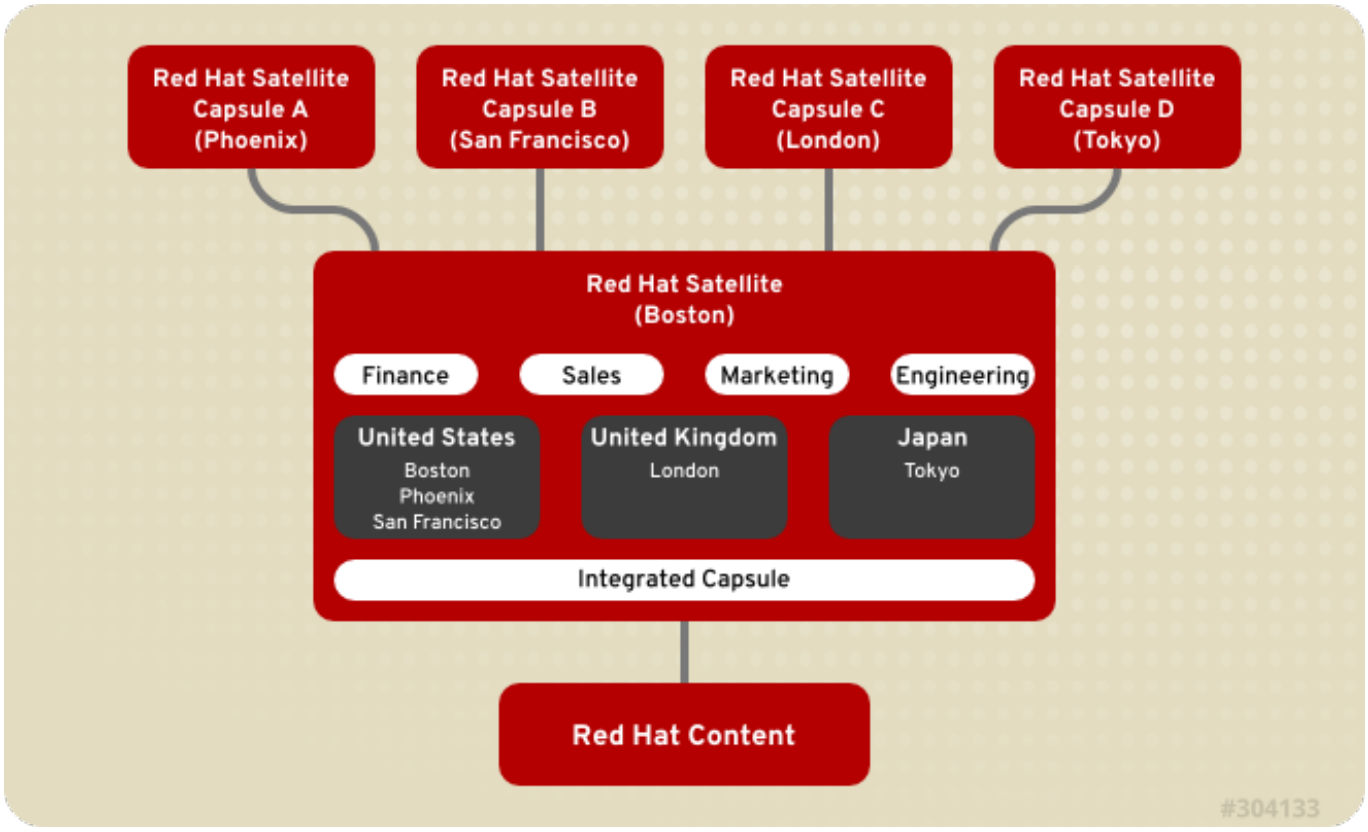


Figure 3.1. Example Topology for Red Hat Satellite 6

The main Satellite server retains the management function, while the content and configuration is synchronized between the main Satellite server and a Satellite Capsule assigned to certain locations.

Report a bug

3.1. Organizations

Organizations divide hosts into logical groups based on ownership, purpose, content, security level, or other divisions.

Multiple organizations can be viewed, created, and managed within the web interface. Software and host entitlements can be allocated across many organizations, and access to those organizations controlled.

Each organization must be created and used by a single Red Hat customer account, however each account can manage multiple organizations. Subscription manifests can only be imported into a single organization and Satellite will not upload a certificate that has already been uploaded into a different organization.

By default, **Red Hat Satellite** will have one organization already created, called ***Default Organization***, which can be modified to suit your own installation, or deleted.

Important

If a new user is not assigned a default organization their access will be limited. To grant the user systems rights, assign them a default organization and have them log out and log back in again.

Report a bug

3.1.1. Creating an Organization

These steps show how to create a new organization.

Procedure 3.1. Creating an Organization

1. Click the **Administer** → **Organizations** menu on the top right hand corner.
2. Click the **New Organization** button.
3. Type in the name of the new organization in the **Name** field.
4. Type in the label of the new organization in the **Label** field.
5. Type in a description of the new organization in the **Description** field.
6. Click the **Submit** button.
7. Select the hosts to assign to the new organization.
   - Click the **Assign All** button to assign all hosts with no organization to the new organization.
   - Click the **Manually Assign** button to manually select and assign the hosts with no organization.
   - Click the **Proceed to Edit** button to skip assigning hosts.

Result:

A new organization is created.

Report a bug

3.1.2. Creating an Organization Debug Certificate

These steps show how to generate and download a debug certificate for an organization. Debug certificates unlock all content from an organization and are required for exporting provisioning templates.

Procedure 3.2. Creating a New Organization Debug Certificate

1. Click the **Administer** → **Organizations** menu on the upper right corner.
2. Select an existing organization from the list on the left.
3. Click the **Generate and Download** button. This generates a debug certificate. Save the certificate in a secure location.

Note

Debug Certificates are automatically generated for provisioning template downloads if they do not already exist in the organization for which they are being downloaded.

Result:

Red Hat Satellite generates a debug certificate and saves it to a location of your choice.

Report a bug

3.1.3. Editing an Organization

Procedure 3.3. Editing an Organization

1. Click the **Administer** → **Organizations** menu on the top right hand corner.
2. Click the name of the organization to be edited.
3. Select the resource to edit from the list on the left.
4. Click the name of the desired items to add them to the **Selected Items** list.
5. Click the **Submit** button.

Result

The organization is updated and saved.

Report a bug

3.1.4. Removing an Organization

Procedure 3.4. Removing an Organization

1. Click the **Administer** → **Organizations** menu on the top right hand corner.
2. Select **Delete** from the drop down menu to the right of the name of the organization you want to remove.
3. An alert box appears:

```
Delete Organization Name?
```

4. Click the **OK** button.

Result

The organization is removed from **Red Hat Satellite**.

Report a bug

3.2. Locations

Locations divide organizations into logical groups based on geographical location.

Each location must be created and used by a single Red Hat customer account, however each account can manage multiple locations and organizations.

By default, **Red Hat Satellite** will have one location already created, called ***Default***, which can be modified to suit your own installation, or deleted.

Important

If a new user is not assigned a default location their access will be limited. To grant the user systems rights, assign them a default location and have them log out and log back in again.

Report a bug

3.2.1. Creating a Location

These steps show how to create a location.

Procedure 3.5. Creating a Location

1. Click the **Administer** → **Locations** menu on the top right hand corner.
2. Click the **New Location** button.
3. Type in the name of the new location in the **Name** field and click the **Submit** button.
4. Select the hosts to assign to the new location.
   - Click the **Assign All** button to assign all hosts with no location to the new location.
   - Click the **Manually Assign** button to manually select and assign the hosts with no location.
   - Click the **Proceed to Edit** button to skip assigning hosts.

Result:

A location is created.

Report a bug

3.2.2. Editing a Location

Procedure 3.6. Editing a Location

1. Click the **Administer** → **Locations** menu on the top right hand corner.

2. Click the name of the location to be edited.
3. Select the resource to edit from the list on the left.
4. Click the name of the desired items to add them to the **Selected Items** list.
5. Click the **Submit** button.

Result

The location is updated and saved.

Report a bug

3.2.3. Removing a Location

These steps show how to remove an existing location.

Procedure 3.7. Removing a Location

1. Click the **Administer** → **Locations** menu on the top right hand corner.
2. Select **Delete** from the drop down menu to the right of the name of the location you want to remove.

   An alert box appears:

   ```
   Delete Location Name
   ```

3. Click the **OK** button.

Result

The location is removed from **Red Hat Satellite**.

Report a bug

3.3. Lifecycle Environments

The application life cycle is divided into *life cycle environments*, which mimic each stage of the life cycle. These life cycle environments are linked in an *environment path*. You can promote content along the environment path to the next life cycle stage when required. For example, if development completes on a particular version of an application, you can promote this version to the testing environment and start development on the next version.

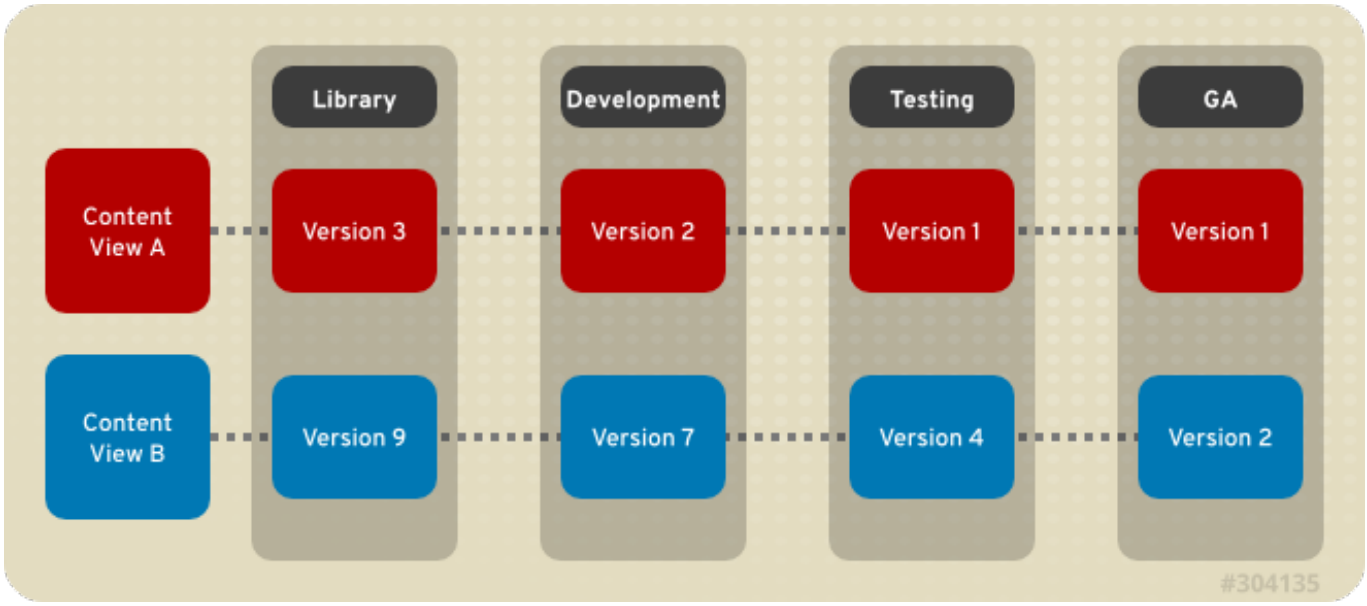


Figure 3.2. An Environment Path Containing Four Environments.

Report a bug

3.3.1. Creating a Lifecycle Environment

This procedure shows how to add a lifecycle environment with **Satellite**.

Procedure 3.8. Creating a Lifecycle Environment

1. Click the **Content** → **Lifecycle Environments** menu.
2. Click the **New Environment Path** button.
3. Type in a name in the **Name** field.
4. Type in a label in the **Label** field.
5. Type in a description in the **Description** field.
6. Click the **Save** button.

A lifecycle environment is created in **Satellite**.

Report a bug

3.3.2. Removing a Lifecycle Environment

These steps show how to remove a lifecycle environment in **Satellite**.

Procedure 3.9. Removing a Lifecycle Environment

1. Click the **Content** → **Lifecycle Environments** menu.
2. Click on the name of the lifecycle environment to be removed.
3. Click on the **Remove Environment** button on the right hand side.
4. An alert box appears:

```
Are you sure you want to remove environment Environment Name?
```

5. Click on the **Remove** button.

Note

Only the last environment can be deleted. If three environments exist in the following order: **Library**, **Dev** and **Prod**, you can delete only the **Prod** environment. The **Library** environment can never be deleted.

Result:

The lifecycle environment is removed.

Report a bug

Chapter 4. Using Content Management

4.1. Using the Red Hat Satellite Content Dashboard

The Satellite Content Dashboard

The dashboard provides a status overview of the subscriptions and hosts currently registered, an overview of promotions and synchronization, and a list of the latest notifications.

Satellite is used to manage entitlements for client machines. Each entitlement provides access to a specified number of certificates. Each certificate grants the right for the client machine to download, update, and receive support for a product.

The dashboard is accessed by clicking the **Monitor → Content Dashboard** menu. The dashboard can be rearranged by clicking on a section title and dragging the section to another position.

Content Host Subscription Status

The **Content Host Subscription Status** gives an overview of the status of the subscriptions currently being managed by **Satellite**. A subscription is a purchased certificate that unlocks access to software, upgrades, and security fixes for hosts.

Table 4.1. Host Subscription States

State	Description	Icon
Invalid Subscriptions	Hosts that have products installed, but have not consumed a subscription. These hosts need attention immediately.	
Insufficient Subscriptions	Hosts that have consumed a subscription and have a valid entitlement, but that are not consuming their full entitlements. These hosts should be monitored to ensure they are configured as expected.	
Current Subscriptions	Hosts that have a valid entitlement and are consuming their full entitlements.	

Latest Notifications

All messages produced by the host are listed in the **Latest Notifications** section. This includes administration information, product and subscription changes, and any errors. Clicking on the cog button displays a drop down menu to change the number of notifications displayed. This can be set to 5 results, 15 results, or 30 results.

This section should be monitored for global notifications sent to all users as well as to pick up any unusual activity or errors.

Sync Overview

An overview of all products or repositories enabled in **Satellite** and their Synchronization status. All products that are in the queue for synchronization, are unsynchronized or have been previously synchronized are listed in the **Sync Overview** section. Click a product name to view the synchronization status.

Clicking on the cog button displays a drop down menu to change the number of notifications displayed. This can be set to 5 results, 15 results, or 30 results.

Host Collections

A list of all Host Collections in **Satellite** and their status, including the number of content hosts in each host collection. Click a host collection name to view that host collection. Clicking on the cog button displays a drop down menu to change the number of notifications displayed. This can be set to 5 results, 15 results, or 30 results.

Current Subscription Totals

An overview of the current subscription totals thats shows the number of active subscriptions, the number of subscriptions that expire in the next 120 days, and the number of subscriptions that have recently expired. Clicking on the number for each type of subscription will show a list of those subscriptions.

Content Views Overview

A list of all Content Views in **Satellite** and their publish status. Clicking on the cog button displays a drop down menu to change the number of notifications displayed. This can be set to 5 results, 15 results, or 30 results.

Errata Overview

A list of all errata in **Satellite**. Clicking on the cog button displays a drop down menu to change the number of notifications displayed. This can be set to 5 results, 15 results, or 30 results.

Report a bug

4.2. Connected Satellite

Red Hat Satellite provides different types of content to subscribed client hosts. Content types include packages, errata updates, kickstart trees, and installation images.

Satellite Server requires a source to provide this content. The content is configured by uploading a *subscription manifest* file to the Satellite. This file can be obtained through the Red Hat Customer Portal, or by contacting Red Hat Support. Manifests provide subscriptions to client hosts through the Red Hat Satellite rather than through Red Hat Network.

This chapter outlines the process of populating your Red Hat Satellite Server, whether it is a connected Red Hat Satellite Server or a disconnected Red Hat Satellite Server, with the content it requires so client hosts can be subscribed to it and receive updates.

Report a bug

4.2.1. Using Red Hat Content Providers

4.2.1.1. Setting up a Manifest

A *subscription manifest* can be obtained through the method below or by contacting Red Hat Support. The manifest is used to set up Red Hat content providers and contains repository information and subscriptions. It is used as a basis of dispensing subscriptions and Red Hat Network (RHN) content to client systems from **Red Hat Satellite**.

Prerequisites

You must meet the following conditions before continuing with this task:

- A Customer Portal user name and password.
- Sufficient subscriptions to add to the manifest.

These steps show how to obtain the subscription manifest from the Customer Portal:

1. Log in to the Customer Portal.
2. Click **Subscriptions** → **Subscriptions Management** → **Subscriptions Management Applications** and then click **Satellite**.
3. On the upper right corner of the Subscriptions Management Applications page, click **Register a Satellite**.
4. Create a name to distinguish your Satellite from the other Satellite systems in your account.
5. Select **6.0** from the drop-down menu as the **Satellite Version**. It is important to select the correct version as each version requires a certain subset of packages.
6. Click **Register**.
7. Click **Attach a subscription**, add the subscriptions required for Red Hat Satellite, and then click **Attach Selected**. See How to generate a certificate for more information.
8. Click **Download manifest** to generate an archive in .zip format that contains the manifest for Red Hat Satellite.

Result:

A subscription manifest is created and downloaded for Red Hat Satellite.

Report a bug

4.2.1.2. Uploading a Subscription Manifest

This section describes how to upload a subscription manifest to an organization. Because subscription manifests are assigned to an organization, ensure you select an organization before you try to upload a subscription manifest. Failing to do so will cause a permission denied error (Error 403).

Procedure 4.1. To Upload Subscription Manifest:

1. Log in to the **Satellite** server.
2. Click **Any Context** → **Any Organization** and select the organization that you want to assign the subscription manifest to.
3. Click **Content** → **Red Hat Subscriptions** and then click **Manage Manifest** at the upper right of the page.
4. In the **Subscription Manifest** section, click **Actions** and under the **Upload New Manifest** subsection, click **Browse**.
5. Select the manifest file to upload, and then click **Upload**.

Report a bug

4.2.1.3. Enabling Red Hat Repositories

The Red Hat manifest file gives access to Red Hat products and repositories, however, since most products have several architectures and product versions, Red Hat Satellite Server allows the Satellite administrators to choose which repositories are required by their organizations. The repositories need to be enabled in the Red Hat Satellite Server to prepare it for synchronization.

These steps show how to enable a Red Hat repository:

Procedure 4.2. Enabling Repositories

1. Click **Content → Red Hat Repositories**.
2. Click on the tab of the type of content to be enabled. The tabs are: **RPMs**, **Source RPMs**, **Debug RPMs**, **Beta**, **ISOs**, **Other**.
3. Expand each Red Hat product to examine the different repository sets available by clicking on the arrow by the product name.
4. Choose which Red Hat repository sets you wish to add. Choosing it will automatically enable that repository for your Red Hat Satellite server.

   For example, a common basic set of subscriptions which contain repositories with the latest packages for Red Hat Enterprise Linux 6 would be:

   a. Red Hat Enterprise Linux 6 Server Kickstart x86_64 6Server Repository

   b. Red Hat Enterprise Linux 6 Server RPMs x86_64 6Server Repository

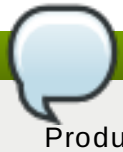
   c. Red Hat Enterprise Linux 6 Server - RH Common RPMs x86_64 Repository

Note

The Red Hat Common repository must be enabled as it provides client systems registered to the Satellite Server katello-agent and puppet packages for both katello-agent and puppet-agent.

Result:

Repositories have been enabled and are ready to be synchronized.



Note

Products in the "Enabling Red Hat Repositories" page are dependent on manifest content. If a required Red Hat repository is not available on this page, consult the manifest or the Red Hat Customer Portal Subscriptions page.

Report a bug

4.2.2. Using Products

4.2.2.1. Creating a Product

These steps show how to create a new product in **Red Hat Satellite**.

Procedure 4.3. Creating a Product

1. Click **Content** → **Products**.
2. Click the **+ New Product** link.
3. Type in the name of the new product in the **Name** field.
4. Type in label for the new product in the **Label** field.
5. Select a GPG key from the **GPG Key** drop down menu.
6. Select a synchronization plan from the **Sync Plan** drop down menu. Alternatively select the **+ New Sync Plan** link to create a new synchronization plan.
7. Type in a description of the new product in the **Description** field.
8. Click the **Save** button to save your new product.

Result:

A new product is created.

Report a bug

4.2.2.2. Adding Repositories to a Product

These steps show how to add repositories to a product in **Red Hat Satellite**.

Procedure 4.4. Adding Repositories to a Product

1. Click **Content** → **Products**.
2. Click the product you wish to add a repository to.
3. Click the **Repositories** subtab.
4. Click the **Create Repository** button.
5. Type in the name of the new repository in the **Name** field.
6. Type in a label for the new repository in the **Label** field.
7. Select the type of the repository from the **Type** drop down menu.
8. Type in the URL of the repository in the **URL** field.
9. Choose whether to publish the repository via HTTP by clicking the **Publish via HTTP** checkbox.
10. Select a GPG key for the repository from the **GPG Key** drop down menu.
11. Click the **Create** button to save your new repository.

Result:

A new repository is added to your product.

Report a bug

4.2.2.3. Using Bulk Actions for Products

This section describes how to use bulk actions to synchronize or remove products in Red Hat Satellite. The procedure described here requires that at least one product be available.

Procedure 4.5. To Perform Tasks on Multiple Products:

1. Click **Content** → **Products**.
2. Select the check box for the products you want to work with.
3. Click **Bulk Actions**.
4. A. To synchronize all selected products, click the **Product Sync** tab and then click **Sync Now**.

   B. To remove all selected products, click **Remove Products** and then click **Remove**.

Updating Synchronization Plans

You can also update the synchronization plans for multiple products at the same time.

- To create a new synchronization plan, click **Create Sync Plan**.
- To remove the synchronization plans from the selected products, click **Unattach Sync Plan**.
- To update the synchronization plans for the selected products, click **Update Sync Plan**.

Report a bug

4.2.2.4. Using Repository Discovery

Repository discovery allows you to search a URL to discover repositories available there to include in a product.

Procedure 4.6. Using Repository Discovery

1. Click the **Content** → **Products** menu.
2. Click the **Repo Discovery** button.
3. Enter the URL where the repositories are located in the **Yum Repo Discovery** field.
4. Click the **Discover** button.
5. A list of the repositories at the URL is displayed under **Results**.
6. Click the **Discovered URLs** check box for the repositories to be added to a product.
7. Click the **Create selected** button.
8. Choose whether to add the repositories to an existing product or create a new product.
   a. To add the repositories to an existing product:
      i. Select the **Existing Product** radio button.
      ii. Select the required product from the drop down menu.

b. To create a new product to add the repositories to:

i. Select the **New Product** radio button.

ii. Enter the **Name** and **Label** for the new product and select a **GPG Key** from the drop down menu.

9. Select the **Serve via HTTP** check box to serve the repository via HTTP.
10. Edit the **Name** and **Label** for the **Selected URLs**.
11. Click the **Create** button.

Result:

The repositories have been discovered and added to a product.

Report a bug

4.2.2.5. Removing a Product

This section describes how to remove products from Red Hat Satellite.

Procedure 4.7. To Remove a Product from Satellite:

1. Click **Content** → **Products**.
2. Select the check box next to the products you want to remove.
3. Click **Bulk Actions** and then click **Remove Products**.
4. Click **Remove** to confirm that you want to remove the products.

Report a bug

4.2.3. Synchronizing Content

Synchronization is the act of coordinating updates between the Red Hat Satellite repositories and the source repositories being used. It is a required step after enabling repositories, in order to populate the Red Hat Satellite with content from the source repositories.

Constant, scheduled synchronization will result in:

- Data integrity between packages
- Updated packages, security fixes, and errata

Satellite's synchronization management capabilities allow organization administrators to create synchronization plans to configure how often a host should look for and install updates. Synchronization plans are then paired with the product repositories to come up with a synchronization schedule that will allow products to be updated at specific intervals that are convenient for the organization's network.

Report a bug

4.2.3.1. Synchronization Status

Important

The manual synchronization of repositories is required after enabling them. It is at this point that the local repository in the Satellite is populated by the required packages.

These steps show how to synchronize products in **Red Hat Satellite**.

Procedure 4.8. Synchronize Products

1. Click **Content** → **Sync Status**. Based on the subscriptions and repositories enabled, the list of product channels available for synchronization is displayed.
2. Click the arrow next to the product name to see available content.
3. Select the content you wish to synchronize.
4. Click the **Synchronize Now** button to starting synchronizing. The status of the synchronization process will appear in the **Result** column. If synchronization is successful, **Sync complete** will appear in the **Result** column. If synchronization failed, **Error syncing** will appear.

Result:

A product is synchronized.

Note

Content synchronization can take a long time. The length of time required is dependent on the speed of disk drives, network connection speed, and the amount of content selected for synchronization.

Report a bug

4.2.3.2. Creating a Synchronization Plan

Regular, frequent synchronization is required to maintain data integrity between packages as well as making sure that packages are updated to the latest security fixes. Red Hat Satellite provides the ability to create scheduled synchronization plans that allow package updates at intervals convenient to the organization.

Procedure 4.9. To Create a Synchronization Plan:

1. Click **Content** → **Sync Plans**.
2. Click the **New Sync Plan** link to create a new synchronization plan.
3. Enter the **Name**, **Description** and other details for the plan.
4. Click **Save** to create the synchronization plan.

Report a bug

4.2.3.3. Applying a Synchronization Schedule

After you have created a synchronization plan, you need to associate products with that plan to create a synchronization schedule. The following procedure describes how to create a synchronization schedule in Red Hat Satellite 6.

Procedure 4.10. To Create a Synchronization Schedule:

1. Click **Content** → **Sync Plans** and select the synchronization plan you want to implement.
2. Click **Products** → **Add** in the synchronization plan main page.
3. Select the check box of the product to associate with the synchronization plan.
4. Click **Add Selected**.

Report a bug

4.3. Disconnected Satellite

In high security environments where hosts are required to function in a closed network, disconnected from the internet, the Red Hat Satellite Server can provision systems with the latest security updates, errata, and packages. This is achieved by using two important components: the **katello-disconnected** utility and a synchronization host.

The diagram below illustrates how a disconnected Satellite is able to keep its content updated even without an internet connection. An intermediary system with an internet connection is needed to act as a synchronization host. This synchronization host is in a separate network from the Satellite server.

The synchronization host imports content from the Red Hat Content Delivery Network (CDN) through pulp. The content is then exported onto a media, such as DVDs, CDs, or external hard drives and transferred to the disconnected Satellite server. The following sections in this chapter will guide you through the whole process.

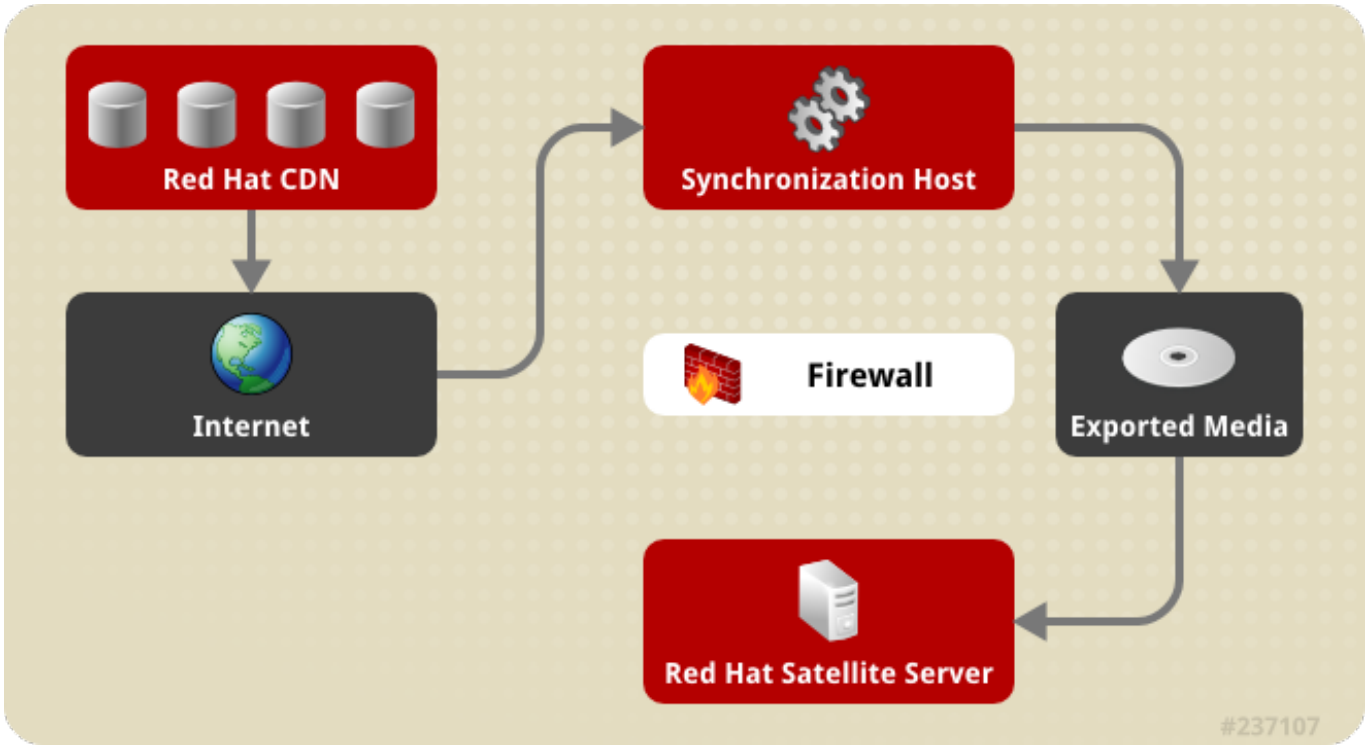


Figure 4.1. Disconnected Satellite

Report a bug

4.3.1. Configuring the Synchronization Host

Prerequisites

To import content from the Red Hat Content Distribution Network (CDN), the synchronization host requires:

- An Internet connection
- Valid Red Hat Network subscriptions
- A valid manifest (See Section 4.2.1.1, “Setting up a Manifest” for instructions on how to obtain one.)

Procedure 4.11. To Configure a Host to Synchronize and Export Content from the Red Hat CDN:

1. Use Red Hat Subscription Manager to register the synchronization host to RHN.
2. List all the available subscriptions to find the correct Red Hat Satellite product to allocate to your system:

```
# subscription-manager list --available --all
```

This command displays output similar to the following:

```
+-------------------------------------------+
    Available Subscriptions
+-------------------------------------------+


ProductName:         Red Hat Satellite
ProductId:           SKU123456
PoolId:              e1730d1f4eaa448397bfd30c8c7f3d334bd8b
Quantity:            10
Multi-Entitlement:   No
Expires:             08/20/2013
MachineType:         physical
```

Note

The SKU and Pool ID depend on the Red Hat Satellite product type that corresponds to your system version and product type.

3. Subscribe to the pool using the following command:

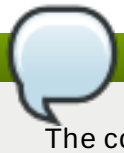
```
# subscription-manager subscribe --pool=Red_Hat_Satellite_Pool_Id
# subscription-manager subscribe --
pool=Red_Hat_Enterprise_Linux_Pool_Id
# subscription-manager subscribe --
pool=Red_Hat_Enterprise_Linux_Software_Collections_Pool_Id
```

4. Disable all existing repositories:

```
# subscription-manager repos --disable "*"
```

5. Enable the Red Hat Satellite and Red Hat Enterprise Linux and Red Hat Software Collections repositories. Ensure the Red Hat Enterprise Linux repository matches the specific version you are using.

```
# subscription-manager repos --enable rhel-6-server-rpms \
--enable rhel-server-rhscl-6-rpms \
--enable rhel-6-server-satellite-6.0-rpms
```



Note

The commands above are based on Red Hat Enterprise Linux 6. If you are using a different version of Red Hat Enterprise Linux, change the repository based on your specific version.

6. Install *katello-utils* and associated RPMs:

```
# yum install python-qpid-qmf python-qpid  qpid-cpp-server-store
katello-utils
```

katello-utils includes the **katello-disconnected** utility that is required to set up repositories for import while *qpid* related packages are necessary for **pulp** configuration.

7. Generate a secret 32-character alphanumeric string for the ***oauth_secret*** entry in the **/etc/pulp/server.conf** file:

```
# tr -dc "[:alnum:]" < /dev/urandom | head -c 32
randomly_generated_value
```

8. In the **/etc/pulp/server.conf**, uncomment the ***[oauth]*** entry and add the randomly generated value from the previous step as the ***oauth_secret*** value:

```
[oauth]
enabled: true
oauth_key: katello
oauth_secret: randomly_generated_value
```

9. Disable authentication in **/etc/qpid/qpidd.conf**:

```
# Configuration file for qpidd. Entries are of the form:
#   name=value
#
# (Note: no spaces on either side of '=').
# Run "qpidd --help" or see "man qpidd" for more details.

auth=no
```

All incoming connections authenticate using the Satellite's default realm.

10. Configure the connection from **katello-disconnected** to Pulp with the previously generated value as your ***--oauth-secret*** option:

```
# katello-disconnected setup --oauth-key=katello --oauth-
secret=randomly_generated_value
```

This places a configuration value in **~/.katello-disconnected**.

11. Configure Pulp on the Synchronization Server:

```
sudo service qpidd start
sudo chkconfig qpidd on
sudo service mongod start
sleep 10
sudo chkconfig mongod on
sudo -u apache pulp-manage-db
sudo service httpd restart
sudo chkconfig pulp_workers on
sudo service pulp_workers start
sudo chkconfig pulp_celerybeat on
sudo service pulp_celerybeat start
sudo chkconfig pulp_resource_manager on
sudo service pulp_resource_manager start
```

12. Import the manifest:

```
# katello-disconnected import -m ./manifest.zip
```

Importing the manifest sets up the list of available repositories to synchronize to based on the subscriptions you selected.

The synchronization host is now ready to synchronize content from the Red Hat CDN.

Report a bug

4.3.2. Synchronizing Content

By default, **katello-disconnected** enables all repositories that are included in the manifest for synchronization. Synchronization time is directly related to the amount of repositories to be synchronized. If the manifest has a large amount of repositories, the synchronization will take time and network resources.

katello-disconnected allows for the synchronization of specific repositories. This section will set up Pulp for synchronizing content.

1. Disable all repositories:

```
# katello-disconnected disable --all
```

katello-disconnected enables all repositories by default.

2. Choose which repositories you wish to sync by listing all available repositories from the manifest:

```
# katello-disconnected list --disabled
```

```
rhel-5-server-debug-rpms-5Server-i386
rhel-5-server-debug-rpms-5Server-ia64
rhel-5-server-debug-rpms-5Server-x86_64
rhel-5-server-debug-rpms-5_7-i386
rhel-5-server-debug-rpms-5_7-ia64
rhel-5-server-debug-rpms-5_7-x86_64
rhel-5-server-debug-rpms-5_8-i386
rhel-5-server-debug-rpms-5_8-ia64
rhel-5-server-debug-rpms-5_8-x86_64
rhel-5-server-debug-rpms-5_9-i386
rhel-5-server-debug-rpms-5_9-ia64
rhel-5-server-debug-rpms-5_9-x86_64
rhel-5-server-isos-5Server-i386
```

3. Enable the chosen repositories for synchronization:

```
# katello-disconnected enable -r rhel-6-server-sam-rpms-6_4-x86_64
```

4. Create the repositories and push them to **Pulp** to allow synchronization:

```
# katello-disconnected configure
```

Note

The configure option for **katello-disconnected** reads the manifest, creates pulp repositories, and generates scripts before synchronization. It needs to be run each time a repository is enabled or disabled.

5. Synchronize the repositories:

```
# katello-disconnected sync
```

You can use the **watch** option to monitor the synchronization process.

```
# katello-disconnected watch
Watching sync... (this may be safely interrupted with Ctrl+C)
running:
rhel-6-server-sam-rpms-6_4-x86_64

running:
rhel-6-server-sam-rpms-6_4-x86_64
...
finished:
rhel-6-server-sam-rpms-6_4-x86_64


Watching finished
```

Content is now synchronized.

Report a bug

4.3.3. Exporting Content

Prerequisites

An external export media such as a CD, DVD, or external hard drive.

The synchronized content needs to be exported to enable importing into the disconnected Red Hat Satellite. To do so:

1. Export the synchronized repositories:

```
# katello-disconnected export -t /var/tmp/export
```

The output will look similar to:

```
# katello-disconnected export -t /var/tmp/export
# katello-disconnected watch
Watching sync... (this may be safely interrupted with Ctrl+C)
running:
rhel-6-server-sam-rpms-6_4-x86_64

finished:
rhel-6-server-sam-rpms-6_4-x86_64
Watching finished
Done watching ...
 Copying content to /var/tmp/export
 Archiving contents of /var/tmp/export into 4600M tar archives.
 NOTE: This may take a while.
tar: Removing leading `/' from member names

Done exporting content, please copy /var/tmp/export/* to your
disconnected host
```

This operation will create the following files in **/var/tmp/export**:

```
# ls /var/tmp/export/
content-export-00 content-export-01 content-export-02
expand_export.sh
```

2. Copy the files from /var/tmp/export into the external media.

Note

If the files are too big for your external media, the files can be copied sequentially in a series of DVDs.

The synchronized content has now been exported and ready for importing to the disconnected Satellite server.

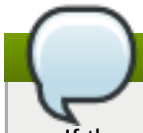
Report a bug

4.3.4. Importing Content to a Disconnected Satellite Server

Prerequisites

Ensure that the directory and filesystem containing the exports has enough space to expand out the archives. For example, if your export is 40G, the disconnected Satellite Server directory and filesystem where you are importing the content will need an extra 40G of space to expand it on the same filesystem.

1. Copy the exported contents of the external media to a directory on the disconnected Satellite.



Note

If the exported content was placed in different external media devices, ensure that you copy all the contents in *one* directory on the disconnected Satellite.

2. Run the **`expand_export.sh`** script copied along with your exported content:

```
# ./expand_export.sh
*** Done expanding archives. ***
```

Warning

The disconnected Satellite needs to have equal to the same space as the exported archive before you expand the exported content. The script will fail otherwise.

The script will expand the archive into a directory.

3. Move the expanded directory tree's content to **`/var/www/html`** on the Satellite Server and restore the SELinux contexts on the moved files:

```
# mv var/tmp/export/* /var/www/html/
# cd /var/www/html
# restorecon -r *
```

Note

The Satellite is now acting as its own CDN with the files located in **`http://localhost/content`**. This is not a requirement. The CDN can be hosted on a different machine inside the same disconnected network as long as it is accessible to the Satellite server via HTTP.

4. Add the CDN address to the Satellite web interface:

    a. Log in to the Satellite web interface.

    b. Click **`Content`** → **`Red Hat Subscriptions`** and then click **`Manage Manifest`**.

    c. On the **`Subscription Manifest`** information screen, scroll to **`Red Hat Provider Details`**. Click the edit icon on the **`Repository URL`** entry and change the entry to the CDN's repository URL.

d. Click **Browse** to choose the manifest file.

e. Click **Upload** to import your manifest.

5. Enable the repositories from the local CDN:

a. Click **Content → Red Hat Repositories**

b. Enable the repositories that were enabled and synchronized in the Synchronizing Content section.

6. Click **Content → Sync Status**.

7. Select the repositories you want to synchronize and click **Synchronize Now**.

Once the synchronize finishes, the disconnected Satellite is now ready to serve the content to client systems.

Report a bug

Chapter 5. Using Content Views

Content views are managed selections of content, which contain one or more repositories (yum / puppet) with optional filtering. These filters can be either inclusive or exclusive, and tailor a system view of content for lifecycle management. They are used to customize content to be made available to client systems.

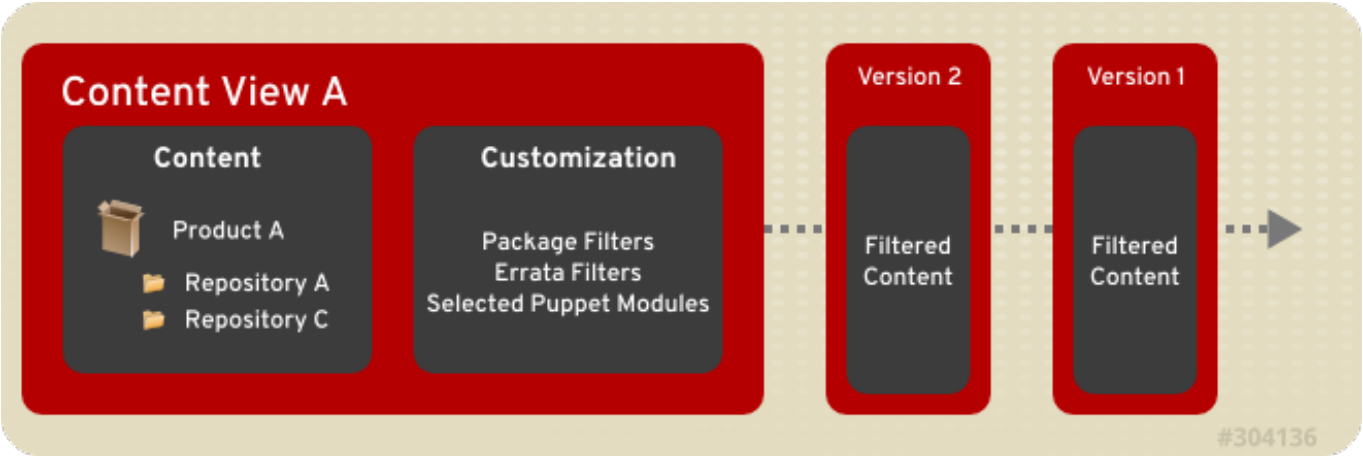


Figure 5.1. This diagram details the creation of new versions of a Content View. These content view versions are promoted along an environment path during the application life cycle.

Published content views are used with life cycle environments.

Report a bug

5.1. Creating a Content View

A user with administrator privileges creates content views for use within the lifecycle environments. To create a content view:

1. Log in as a Satellite administrator.
2. Click **Content → Content Views**.
3. Click **Create New View**.
4. Fill in the following fields:
   - **Name**
   - **Label** - this field is automatically populated when the **Name** field is filled out.
   - **Description**
5. Select the **Composite View** check box to combine a series of published content views into one and choose which content view.

Note

If you select **Composite View** it will override any filtering and allow you to choose a group of published content views and bundle those views into a composite one.

6. Click **Save**.

Report a bug

5.2. Adding Repositories to the Content View

A Repository provides storage for a collection of content. For example, a YUM repository or a Puppet repository. To associate a repository with a specific content view:

1. Click **Content** → **Content View** and choose the Content View to add repositories to.
2. Click **Content** and select **Repositories** from the drop down menu. From the submenu, click **Add**.
3. Select the repositories to add. Once all the intended repositories have been selected, click **+Add Repositories**.

Repositories have now been added to the Content View.

Report a bug

5.3. Filtering Content

Filters are created to prevent packages from being promoted to subsequent environments. Package names or regular expressions are added to the filter to create the rules to blacklist packages and the filter is then associated to entire products or individual repositories within any product.

Report a bug

5.3.1. Creating a Filter

These steps show how to create a filter.

Procedure 5.1. Creating a Filter

1. Click **Content** → **Content Views**.
2. Select the Content View you wish to filter.
3. Click the **Content** → **Filters** subtab.
4. Click on the **+New Filter** button.
5. Type in the name of the new filter in the **Name** field.
6. Choose a content type from the **Content Type** drop down menu.
7. Choose whether the filter includes or excludes the selected content type by selecting the **Type** drop down menu.

8. Optionally, enter a description in the **Description** field.
9. Click the **Save** button to save your new filter.

Result:

A filter is created.

Report a bug

5.3.2. Adding Content to a Filter

Prerequisites

Requires a created Filter.

1. Click **Content → Content Views**.
2. Select the Content View you wish to filter.
3. Click the **Content → Filters** subtab.
4. Click a created package filter's name. Depending on the type of filter selected, the readout will be different.
    a. If the filter is made for Packages:
        i. Enter a package name and select a **Detail** value from the dropdown menu. Click the **+Add** button to add the package to the filter.
        ii. Under the **Affected Repositories** subtab, choose whether the filter will affect all or a subset of repositories. If you choose a subset of repositories, then choose the desired repositories and click the **Update Repositories** button.
    b. If the filter is made for package groups:
        i. Click on the **Add** subtab, and choose the desired package group. Click the **+Add Package Group** button.
        ii. Under the **Affected Repositories** subtab, choose whether the filter will affect all or a subset of repositories. If you choose a subset of repositories, then choose the desired repositories and click the **Update Repositories** button.
    c. If the filter is made for Errata:
        i. Click on the **Add** subtab. Check the desired boxes for the Errata type, whether is be **Security**, **Enhancement**, or **Bugfix**. Then choose a start date and end date. Click the **+Add Errata** button.
        ii. Under the **Affected Repositories** subtab, choose whether the filter will affect all or a subset of repositories. If you choose a subset of repositories, then choose the desired repositories and click the **Update Repositories** button.
    d. If the filter is made for Errata - Date and Type:
        i. Under the **Erratum Date Range** subtab, check the desired boxes for the Errata type, whether is be **Security**, **Enhancement**, or **Bugfix**. Then choose

a start date and end date. Click the **Save** button.

ii. Under the **Affected Repositories** subtab, choose whether the filter will affect all or a subset of repositories. If you choose a subset of repositories, then choose the desired repositories and click the **Update Repositories** button.

5. Click the **Publish New Version** button. Enter a comment if desired, then click the **Save** button.

Result

Packages are added to the filter.

Report a bug

5.3.3. Removing Content from a Filter

Prerequisites

Requires a created Filter.

1. Click **Content** → **Content Views**.
2. Select the Content View you wish to filter.
3. Click the **Content** → **Filters** subtab.
4. Click a created package filter's name. Depending on the type of filter selected, the readout will be different.
    a. If the filter is made for Packages:
        i. Click the **Packages** subtab then click the **Package Name** checkbox next to the package to remove. Click the **Remove Packages** button to remove the package from the filter.
        ii. Under the **Affected Repositories** subtab, choose whether the filter will affect all or a subset of repositories. If you choose a subset of repositories, then choose the desired repositories and click the **Update Repositories** button.
    b. If the filter is made for package groups:
        i. Click the **List/Remove** subtab then click the **Name** checkbox next to the package group to remove. Click the **Remove Package Group** button to remove the package group from the filter.
        ii. Under the **Affected Repositories** subtab, choose whether the filter will affect all or a subset of repositories. If you choose a subset of repositories, then choose the desired repositories and click the **Update Repositories** button.
    c. If the filter is made for Errata:
        i. Click the **List/Remove** subtab then click the **Errata ID** checkbox next to the errata to remove. Click the **Remove Errata** button to remove the errata from the filter.

ii. Under the **Affected Repositories** subtab, choose whether the filter will affect all or a subset of repositories. If you choose a subset of repositories, then choose the desired repositories and click the **Update Repositories** button.

d. If the filter is made for Errata - Date and Type:

i. Under the **Erratum Date Range** subtab, check the desired boxes for the Errata type, whether is be **Security**, **Enhancement**, or **Bugfix**. Then edit the start date and end date. Click the **Save** button.

ii. Under the **Affected Repositories** subtab, choose whether the filter will affect all or a subset of repositories. If you choose a subset of repositories, then choose the desired repositories and click the **Update Repositories** button.

5. Click the **Publish New Version** button. Enter a comment if desired, then click the **Save** button.

Result

Packages are removed from the filter.

Report a bug

5.3.4. Removing a Filter

These steps show how to remove a filter.

Procedure 5.2. Remove a Filter

1. Click **Content** → **Content Views**.
2. Select the Content View you wish to filter.
3. Click the **Content** → **Filters** subtab.
4. Click the checkbox next to the name of the package filter you wish to remove.
5. Click the **Remove Filters** button.

Result:

A filter is removed.

Report a bug

5.4. Publishing a Content View

Once a content view has been created, it needs to be published in order for it to be visible and usable by hosts. Before publishing the content view definition, make sure that the content view definition has the necessary products, repositories and filters.

To publish a content view definition:

1. Click **Content** → **Content Views**.
2. Click on the content view to be published.

3. Click the **Publish New Version** button.
4. Fill in a comment.
5. Click the **Save** button.

A published content view is now available.

Report a bug

Chapter 6. Searching for Content

6.1. Basic Content Search

Content Search allows users to look for specific information about content views, products, repositories, or errata.

Procedure 6.1. To Perform a Content Search:

1. Click **Content** → **Content Search**.
2. Select either **Content Views**, **Products**, **Repositories**, **Packages**, **Errata**, or **Puppet Modules** from the **Content** drop-down menu.
3. Enter the name of the content view, product, repository, package, errata, or puppet module you are searching for in the **Products** field, and click **Search**.

Report a bug

6.2. Content Comparison across Environments

You can compare content across different environments using the *content search* feature.

Procedure 6.2. To Search for and Compare Content Across Different Environments:

1. Click **Content** → **Content Search**.
2. Select the entity type that you want to compare from the **Content** drop-down menu.
3. Enter the name of the entity in the **Products** field and click **Add**.
4. On the right panel, move your cursor over the "plus" (**+**) icon, select the environments you want to compare, and click **Search**.
5. Select either **Union**, **Intersection**, or **Difference** from the **View** drop-down menu to filter your results.

Report a bug

Chapter 7. Configuring Activation Keys

Activation Keys are preset keys used when registering the host and define:

- Which lifecycle environment the host should be placed in.
- Which host collection the host should be assigned to.
- Which organization the host should be a part of.
- Whether to use a provisioning template for the host.
- Setting up a subscription usage limit for the host.
- Assigning a specific subscription to the host.

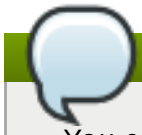
Report a bug

7.1. Creating an Activation Key

This section describes how to create an activation key.

Procedure 7.1. To Create an Activation Key:

1. Click **Content** → **Activation Keys**.
2. Click **New Activation Key**.
3. Enter the required details for the activation key in the relevant fields.
4. Clear the **Unlimited** check box if the activation key is to be used with limitations. Type the usage limit in the **Limit** field.
5. Enter a suitable description in the **Description** field.
6. Select the **Environment** and **Content View** that this key should apply to.
7. Click **Save** to create the activation key.



Note

You can change the activation key details on the **Details** tab of the Activation Key.

Report a bug

7.2. Removing an Activation Key

This section describes how to remove an activation key.

Procedure 7.2. To Remove an Activation Key

1. Click **Content** → **Activation Keys**.
2. Click the activation key name that you want to remove.

3. In the upper right of the **Activation Key** detail panel, click **Remove**.
4. In the alert box, click **Remove** to confirm that you want to remove the key.

Report a bug

7.3. Adding Objects to Activation Keys

This section describes how to add different types of objects to activation keys.

Report a bug

7.3.1. Adding Subscriptions to an Activation Key

This section describes how to add subscriptions to an activation key.

Procedure 7.3. To Add a Subscription to an Activation Key:

1. Click **Content** → **Activation Keys**.
2. Click the name of the activation key that you want to add subscriptions to.
3. Click the **Subscriptions** → **Add**.
4. From the list of available subscriptions, select the subscriptions you want to add and then click **Add Selected**.

Report a bug

7.3.2. Adding Host Collections to an Activation Key

These steps show how to add host collections to an activation key.

Procedure 7.4. To Add Host Collections to an Activation Key:

1. Click **Content** → **Activation Keys**.
2. Click the activation key that you want to add a host collection to.
3. Click **Host Collections** and then click **Add** to display the list of available host collections.
4. Select the host collections you want to add, and then click **Add Selected** to add the host collections to the activation key.

Note

After you have added the host collections to the activation key, they no longer appear in the list of available collections. To view the host collections that have been added to an activation key, click **List/Remove**.

Report a bug

7.4. Removing Objects from Activation Keys

7.4.1. Removing Subscriptions from an Activation Key

These steps show how to remove subscriptions from an activation key.

Procedure 7.5. Remove Subscriptions from an Activation Key

1. Click **Content** → **Activation Keys**.
2. A list of activation keys is displayed. Click the activation key you wish to remove subscriptions from.
3. Click the **Subscriptions** subtab.
4. A list of subscriptions is displayed. Select the subscriptions you wish to remove.
5. Click the **Remove Selected** button to remove subscriptions from the activation key.

Result:

Subscriptions are removed from your activation key.

Report a bug

7.4.2. Removing Host Collections from an Activation Key

These steps show how to remove host collections from an activation key.

Procedure 7.6. Remove Host Collections from the Activation Key

1. Click **Content** → **Activation Keys**.
2. A list of activation keys is displayed. Click the activation key you wish to remove host collections from.
3. Click the **Host Collections** subtab.
4. A list of host collections attached to the Activation Key is displayed. Tick the checkbox of the host collections you wish to remove.
5. Click the **Remove Selected** button to remove host collections from the activation key.

Result:

Host collections are removed from your activation key.

Report a bug

Chapter 8. Configuring GPG Keys

GPG keys allow you to add your existing GPG keys to Red Hat Satellite Server products and repositories to enable pairing with your repositories.

Report a bug

8.1. Creating a GPG Key

This section describes how to add a GPG key to Red Hat Satellite.

Procedure 8.1. To Add a GPG Key to Satellite:

1. Click **Content** → **GPG Keys** and then click **New GPG Key**.
2. Enter a name for the GPG key in the **Name** field.
3. Either upload the GPG key file or paste the GPG key contents into the text box.
4. Click **Save** to add the GPG key to Satellite.

Report a bug

8.2. Removing a GPG Key

This section describes how to remove a GPG from Red Hat Satellite.

Procedure 8.2. To Remove a GPG Key:

1. Click **Content** → **GPG Keys**.
2. Click the GPG key that you want to remove, and then click **Remove GPG Key**.
3. In the confirmation box, click **Remove** to confirm that you want to remove the selected key.

Report a bug

Chapter 9. Configuring the Provisioning Environment

9.1. Host Groups

A host group defines a set of default values that hosts inherit when placed in that group. Hosts can belong to only one host group, but host groups can be nested in hierarchies. you can create a "base" or "parent" host group that represents all hosts in your organization, and then create nested or "child" host groups under that parent to provide specific settings.

Report a bug

9.1.1. Creating a Host Group

This section describes how to add a host group.

Procedure 9.1. To Add a Host Group to Satellite:

1. Click **Configure** → **Host Groups** and then click **New Host Group**.
2. Enter the required details for the Host Group, and then click **Submit**.

Host Group Attributes

The following table describes the attributes that apply to Satellite Host Groups.

Table 9.1. Table of Host Group Attributes

Submenu	Options	Description
Host Group	Parent	The parent Host Group for the new Host Group.
	Name	The name of the Host Group.
	Lifecycle Environment	The environment containing this Host Group.
	Puppet CA	The Red Hat Satellite Capsule Server to use for the Puppet CA server.
	Puppet Master	The Red Hat Satellite Capsule Server to use as the Puppet Master.
Puppet Classes	Included Classes	The Puppet Classes included with the Host Group.
	Available Classes	The Puppet Classes available to use with the Host Group.
Network	Domain	The domain for hosts in the Host Group.
	Subnet	The subnet for hosts in the Host Group.
Operating System	Architecture	The default architecture for systems in the Host Group.
	Operating Systems	The default operating system for systems in the Host Group.
	Media	The location of the installation media for the operating system.
	Partition Table	A file system partition layout for the operating system installation.
	Root Password	The root password for the operating system.
Parameters	Add Parameter	Provides a Name and Value pair to set parameters for the Host Group.
Organizations	Organizations	The organizations that own this host group.

Submenu	Options	Description
Activation Keys	Content Environment	Defines the activation keys made available in templates as `@host.params['kt_activation_keys']`.

Report a bug

9.2. Parameters

Red Hat Satellite parameters define key-value pairs to use when provisioning hosts. These are similar to Puppet's concept of a default scope parameter. You can define parameters when setting up a host with Puppet and also define a hierarchy of parameter inheritance.

The following parameter hierarchy applies:

Global Parameters

The default parameter that applies to every host in Satellite. Configured in **Configure** → **Global parameters**.

Domain Parameters

Parameters that affect all hosts in a given domain. Domain parameters override Global parameters. Configured in **Infrastructure** → **Domains**.

Host Group Parameters

Parameters that affect all hosts in the Host Group. Host Group parameters override both Global and Domain parameters. Configured in **Configure** → **Host Groups**.

Host Parameters

Parameters that affect a specific host. All previous inherited parameters are visible on the Parameters subtab and can be overridden. Configured in **Hosts** → **All hosts** → **[choose_a_host]** → **Parameters** or **Hosts** → **Content Hosts** → **[choose_a_host]** → **Parameters**.

Different types of parameters also exist:

Simple Parameters

A basic parameter that defines a relationship between a key and value pair.

Smart Parameters

A complex parameter that defines a value for a key but allows conditional arguments, validation, and overrides for specific object types.

Parameterized Classes

Parameters for classes imported from a Puppet Master.

Important

Ensure you enable parameterized class support. Navigate to **Administer** → **Settings**, select the **Puppet** tab, and ensure the **Parametrized_Classes_in_ENC** is set to **True**.

Report a bug

9.2.1. Creating a Global Simple Parameter

This procedure shows how to add a new global parameter to Satellite.

1. Click **Configure** → **Global Parameters**.
2. Click the **New Parameter** button.
3. Type a **Name** for the parameter's key.
4. Type a **Value** for the parameter.
5. Click the **Submit** button.

Satellite creates a new simple parameter for global use.

Report a bug

9.2.2. Creating a Puppet Class

This procedure shows how to add a new Puppet Class to Satellite.

1. Click **Configure** → **Puppet Classes**.
2. Click the **New Puppet class** button.
3. Type a **Name** for the Puppet Class.
4. Type a **Puppet Environments** for the Puppet Class.
5. Select one or more **Host groups** that own the Puppet Class.
6. Click the **Submit** button.

Satellite creates a Puppet Class.

Report a bug

9.2.3. Configuring Smart Parameters

The following procedure configures smart parameters in a Puppet class.

Procedure 9.2. To configure smart parameters

1. Click **Configure** → **Puppet Classes**.
2. Select a class from the list.
3. Click the **Smart Variables** tab. This displays a new screen. The left section contains a list of possible parameters the class supports. The right section contains the configuration options for the parameter selected. Click the **Add Variable** to add a new parameter. Otherwise, select a parameter from the left-hand list.
4. Type a name for the **Parameter** field.
5. Edit the **Description** text box to add any plain text notes.

6. Select the **Parameter type** of data to pass. This is most commonly a string, but other data types are supported.
7. Type a **Default Value** for the parameter.
8. Use the **Optional Input Validator** section to restrict the allowed values for the parameter. Choose a **Validator type** (either a **list** of comma separated values or a regular expression, **regexp**) and input the allows values or regular expression code in the **Validator rule** field.
9. The **Override Value For Specific Hosts** section at the bottom of the page provides options for overriding values based upon conditional arguments known as **Matchers**. Define the **Order** that the host values resolve, then click **Add Matcher-Value** to add your conditional argument.

   For example, if desired value of the parameter is **test** for any host with a fully qualified domain name of **www.example.com**, then specify the **Match** as **fqdn=www.example.com** and the **Value** as **test**.
10. Click **Submit** to save your changes.

Result:

Satellite configures the smart parameter.

Report a bug

9.2.4. Importing Parameterized Classes from a Puppet Master

The following procedure imports parameterized classes from your Puppet Master.

Procedure 9.3. To import parameterized classes

Note

The import of parameterized classes happens automatically if your puppet modules are managed via a Product and a Content View.

1. Click **Configure** → **Puppet Classes**.
2. Click **Import from *Host Name*** to import parameterized classes from your Puppet Master.
3. The **Puppet Classes** page displays with the new classes listed.

Result:

Satellite imports the Puppet Master's parameterized classes.

Report a bug

9.2.5. Configuring Parameterized Classes

The following procedure configures parameterized classes.

Procedure 9.4. To configure parameterized classes

1. Click **Configure → Puppet Classes**.
2. Select a class from the list.
3. Click the **Smart Class Parameter** tab. This displays a new screen. The left section contains a list of possible parameters the class supports. The right section contains the configuration options for the parameter selected.
4. Select the a parameter from the left-hand list.
5. Edit the **Description** text box to add any plain text notes.
6. Click the **Override** checkbox to allow Satellite control over this variable. If the checkbox is not selected, Satellite does not pass this variable to Puppet.
7. Select the **Parameter type** of data to pass. This is most commonly a string, but other data types are supported.
8. Type a **Default Value** for the parameter.
9. The **Override Value For Specific Hosts** section at the bottom of the page provides options for overriding values based upon conditional arguments known as **Matchers**. Define the **Order** that the host values resolve, then click **Add Matcher-Value** to add your conditional argument.

   For example, if desired value of the parameter is **test** for any host with a fully qualified domain name of **www.example.com**, then specify the **Match** as **fqdn=www.example.com** and the **Value** as **test**.
10. Click **Submit** to save your changes.

Result:

Satellite configures the parameters for the class.

Report a bug

9.3. Configuring Provisioning Settings

9.3.1. Domains

Satellite has the ability to assign domain names with Red Hat Satellite Capsule Server DNS. This provides users with a means to group and name hosts within a particular domain.

Report a bug

9.3.1.1. Creating a Domain

This procedure shows how to add a Domain.

Procedure 9.5. Creating a Domain

1. Click **Infrastructure → Domains**.
2. Click the **New Domain** button.
3. Type a **Name** for the Domain. This is the DNS domain name.

4. Type a **Description** for the Domain.
5. Choose a DNS-enabled Red Hat Satellite Capsule Server from the **DNS Proxy** selection box.
6. Click the **Locations** tab and click the desired locations to add them to the **Selected Items** list.
7. Click the **Organizations** tab and click the desired organizations to add them to the **Selected Items** list.

Important

Ensure that the Locations and Organizations are configured as they will help with future debugging.

8. Click the **Submit** button.

Satellite creates a Domain and registers it on the DNS server configured with the selected Smart Proxy.

Report a bug

9.3.2. Subnets

Satellite has the ability to create networks for groups of systems. Subnets use standard IP address settings to define the network and use the Red Hat Satellite Capsule Server's DHCP features to assign IP addresses to systems within the subnet.

Report a bug

9.3.2.1. Creating a Subnet

The following procedure shows how to create a Subnet:

Procedure 9.6. Creating a Subnet

1. Click **Infrastructure** → **Subnets**.
2. Click the **New Subnet** button.
3. Type a **Name** for the Subnet.
4. Type the IP address for the Subnet into the **Network** box.
5. Type the mask for the network's IP address into the **Network mask** box.
6. Type the **Gateway address** for the Subnet.
7. Type the **Primary DNS server** and **Secondary DNS server** for the Subnet.
8. Define the IP assignment range with the **Start of IP range** and **End of IP range** fields.
9. Define the **VLAN ID** for the subnet.
10. Select the applicable domain for the subnet from the **Domains** tab.

11. Click the **Capsules** tab, and select a Red Hat Satellite Capsule Server for the **DHCP Proxy**, **TFTP Proxy**, and **DNS Proxy** services.
12. Click the **Locations** tab and click the desired locations to add them to the **Selected Items** list.
13. Click the **Organizations** tab and click the desired organizations to add them to the **Selected Items** list.

Important

Ensure that the Locations and Organizations are configured as they will help with future debugging.

14. Click the **Submit** button.

Satellite creates a Subnet.

Report a bug

9.3.3. Architectures

An architecture in Satellite represents a logical grouping of hosts and operating systems. Architectures are created by Satellite automatically when hosts check in with Puppet. However, none exist with a default installation and require creation.

Report a bug

9.3.3.1. Creating an Architecture

This procedure shows how to add an Architecture.

Procedure 9.7. Creating an Architecture

1. Click **Hosts** → **Architectures**.
2. Click the **New Architecture** button.
3. Type a **Name** for the Architecture.
4. Select any **Operating Systems** that include this Architecture. If none are available, you can create and assign them under **Hosts** → **Operating Systems**.
5. Click the **Submit** button.

Satellite creates a Architecture.

Report a bug

9.3.4. Compute Resources

Compute resources are hardware abstractions from virtualization and cloud providers. Satellite uses compute resources to provision virtual machines. Supported private providers include Red Hat Enterprise Virtualization, oVirt, OpenStack, and VMware. Supported public cloud providers include EC2 and Rackspace.

Report a bug

9.3.4.1. Creating a Compute Resource

This procedure shows how to add a Compute Resource.

Procedure 9.8. Creating a Compute Resource

1. Click **Infrastructure** → **Compute Resources**.
2. Click the **New Compute Resource** button.
3. Type a **Name** for the Compute Resource.
4. Select a **Provider** type.
5. Optionally, enter a **Description**.
6. Depending on the provider type chosen, the next few fields ask for authentication and datacenter details. Refer to the following table for more information about each provider type.

Table 9.2. Provider Settings

Type	Description
RHEV	Suits Red Hat Enterprise Virtualization environments. Requires a **URL** to the Manager API, a valid **User** and **Password**, and a **Datacenter** on the system to abstract compute resources. Click the **Load Datacenters** button to populate the selection box.
EC2	Suits Amazon EC2 public cloud accounts. Requires the **Access Key** and **Secret Key** available from any valid Amazon EC2 account. Requires a **Region** to act as a Datacenter for resource abstraction. Click the **Load Regions** button to populate the selection box.
VMware	Suits VMware-based environments. Requires the hostname of the **VCenter/Server**, a valid VMware **User** and **Password**, and a **Datacenter** to abstract compute resources. Click the **Load Datacenters** button to populate the selection box.
OpenStack	Suits OpenStack-based environments. Requires the **URL** of the OpenStack server, a valid OpenStack **User** and **Password**, and a **Tenant** to abstract compute resources. Click the **Load Tenants** button to populate the selection box.

Type	Description
Rackspace	Suits Rackspace public cloud accounts. Requires the **URL** of the Rackspace API, a valid Rackspace **User** and **API Key**, and a **Region** to abstract compute resources. Click the **Test Connection** button to make sure your connection to the chosen region is valid.

7. Click the **Locations** tab and click the desired locations to add them to the **Selected Items** list.
8. Click the **Organizations** tab and click the desired organizations to add them to the **Selected Items** list.

Important

Ensure that the Locations and Organizations are configured as they will help with future debugging.

9. Click the **Submit** button.

Satellite creates a Compute Resource.

Report a bug

9.3.5. Hardware Models

Hardware models help run unattended Solaris installations. For Solaris SPARC-based machines, users define the CPU and Vendor information, while other architectures do not need to do so.

Report a bug

9.3.5.1. Creating a Hardware Model

This procedure shows how to add a Hardware Model.

Procedure 9.9. Creating a Hardware Model

1. Click **Hosts** → **Hardware Models**.
2. Click the **New Model** button.
3. Type a **Name** for the Hardware Model.
4. For Sparc Solaris builds, enter the CPU **Hardware model** and **Vendor class**. Other architectures do not require values in these fields.
5. Enter a description of the Hardware Model in the **Information** textbox.
6. Click the **Submit** button.

Satellite creates a Hardware Model.

Report a bug

9.3.6. Installation Media

Red Hat Satellite uses installation media (ISO images) as content for kickstart trees and new host installations.

Report a bug

9.3.6.1. Adding Installation Media

This procedure shows how to add new Installation Media to Satellite.

1. Click **Hosts** → **Installation Media**.
2. Click the **New Installation Medium** button.
3. Type a **Name** for the Installation Media.
4. Type a **Path** to the Installation Medium. Options include either a URL or a valid NFS server.
5. Select an **Operating System Family** to define the Installation Media's type.
6. Click the **Locations** tab and click the desired locations to add them to the **Selected Items** list.
7. Click the **Organizations** tab and click the desired organizations to add them to the **Selected Items** list.

Important

Ensure that the Locations and Organizations are configured as they will help with future debugging.

8. Click the **Submit** button.

Satellite adds the new Installation Media to its collection.

Report a bug

9.3.7. Operating Systems

Operating Systems define combinations of installation methods and media and are grouped within families. As a default, Red Hat Satellite uses a **RedHat** family. Families allow Satellite to change certain behaviors when provisioning hosts.

Report a bug

9.3.7.1. Adding an Operating System

This procedure shows how to add a Operating System to Satellite.

Procedure 9.10. Adding an Operating System

1. Click **Hosts** → **Operating Systems**.
2. Click the **New Operating system** button.

3. Type a **Name** for the Operating System.
4. Define the **Major Version** of the Operating System.
5. Define the **Minor Version** of the Operating System.
6. Select the **OS Family** to define the Operating System type.
7. Select the **Architectures** from the list of available Architectures. If none are available, create and assign them under **Hosts** → **Architectures**.
8. Click the **Partition tables** tab, then add the applicable file system layouts from the list.
9. Click the **Installation Media** tab, then add the applicable file system layouts from the list.
10. Click the **Submit** button.

Satellite adds the Operating System to its collection.

Report a bug

9.3.8. Partition Tables

Partition tables define the partitions and file system layout for new installations when provisioning systems. Satellite users specify the host's disk layout as an explicit sequence of partitions or use a dynamic disk layout script.

Report a bug

9.3.8.1. Defining a New Partition Table

This procedure shows how to define a new Partition Table for new installations.

1. Click **Hosts** → **Partition Tables**.
2. Click the **New Partition Table** button.
3. Type a **Name** for the partition table.
4. Enter the **Layout** for the Partition Table. The **Layout** textbox also accepts dynamic disk partitioning scripts.
5. Select the operating system from the **OS Family** tab to define the Operating System type for the partitions.
6. Click the **Submit** button.

Satellite adds a new Partition Table to its collection.

Report a bug

9.3.9. Provisioning Templates

Provisioning templates provide the systematic means to run unattended installations. Provisioning templates can be executed via several methods including bash scripts, kickstart scripts, and PXE-based installations.

Report a bug

9.3.9.1. Creating a Provisioning Template

This procedure shows how to create a Provisioning Template.

Procedure 9.11. Creating a Provisioning Template

1. Click **Hosts** → **Provisioning Templates**.
2. Click the **New Template** button.
3. Type a **Name** for the template.
4. Enter your template in the **Template editor** field. Alternatively, upload your template with the **Template** file browser below the **Template editor** textbox. This replaces the content in the **Template editor** field with the content of your chosen file.
5. Enter a comment in the **Audit Comment** field. Satellite adds the comment to the template history to track changes. View the template history under the **History** tab.
6. Click the **Type** tab, then select **Snippet** to store the template code without defining it as particular script or template type, or select the type from the **Type** dropdown menu.
7. Select the **Association** tab to associate the template to Hostgroups, Environments and Operating Systems. Select the operating systems from the **Applicable Operating Systems** list. Click the **Add Combination** button and select a **Hostgroup** and **Environment** to limit the template's use.
8. Click the **Submit** button.

Satellite adds the Provisioning Template to its collection.

Report a bug

Chapter 10. Configuring Hosts

In **Red Hat Satellite**, hosts are client systems which have **Red Hat Subscription Manager** installed. **Red Hat Subscription Manager** sends updates to **Red Hat Satellite** and **Red Hat Satellite** provides updates to these client systems.

Hosts must be registered in order to be managed. Once a host has been registered, it can be viewed and edited in the **Hosts** tab. This enables a user to add and manage subscriptions, add and remove software packages, and apply updates.

Report a bug

10.1. Creating a Host

The following procedure describes how to create a host in Red Hat Satellite.

Procedure 10.1. To Create a Host:

1. Click **Hosts** → **New Host** to open the **New Host** page.
2. On the **Host** tab, enter the required details.
3. On the **Network** tab, enter the **Domain** and **Realm** details. This automatically updates the **Subnet** list with a selection of suitable subnets.
4. Enter the **Primary Interface** details. You can specify an IP address or click **IP address auto-suggest** to automatically select an address.
5. On the **Operating System** tab, enter the required details. You can select a partition table from the drop-down list or enter a custom partition table in the **Custom partition table** field. You cannot specify both.
6. On the **Parameters** tab, click **Add Parameter** to add any required parameters. This includes all Puppet Class Parameters and Host Parameters associated with the host.
7. On the **Additional Information** tab, enter any required information.
8. Click **Submit** to complete your provisioning request.

Report a bug

10.2. Configuring Hosts for Registration

Red Hat Enterprise Linux hosts register to Red Hat Network (RHN) by default. You need to update each host configuration so that they register to and update from the correct Red Hat Satellite Server. Address the following requirements before proceeding:

- On the Red Hat Satellite Server, ensure that the date and time are correct and synchronized with the client.
- On each client system, address the following requirements:
  - Ensure that the date and time are correct and synchronized with the server.
  - Enable **ntpd** or a similar time synchronization tool in all virtual environments:

```
# chkconfig ntpd on; service ntpd start
```

- Install the latest version of Red Hat Subscription Manager:

```
# yum install subscription-manager
```

- Update *yum* to the latest version.

```
# yum update yum
```

Important

In Red Hat Enterprise Linux 6.1, clients with earlier versions of *yum*, specifically yum-3.2.29-17.el6.noarch, are unable to obtain content from the Red Hat Satellite.

Report a bug

10.3. Configuration Options

10.3.1. Automated Configuration

These steps show how to automatically configure your client system to register to **Red Hat Satellite**.

1. Take note of the **Red Hat Satellite** hostname or the fully qualified domain name (fqdn).
2. Open a terminal console and login as root on the command line.
3. Download and install a copy of the CA Certificate for **Red Hat Satellite**:

```
yum -y --nogpgcheck install http://[hostname]/pub/katello-ca-consumer-[hostname]-1.0-1.noarch.rpm
```

Important

yum in Red Hat Enterprise Linux 5 does not support installation via HTTP. If registering a Red Hat Enterprise Linux 5 client, download the RPM package first and then run **yum** on the package. For example:

```
# wget http://[hostname]/pub/katello-ca-consumer-[hostname]-1.0-1.noarch.rpm
# yum -y --nogpgcheck install katello-ca-consumer-[host]-1.0-1.noarch.rpm
```

Note

katello-ca-consumer-latest.noarch.rpm is an additional katello-ca-consumer rpm available that does not contain the server's hostname. This rpm will always reflect the most updated version of the package. Both serve the same purpose.

Result

The client system is now ready to register to **Red Hat Satellite**.

Report a bug

10.3.2. Manual Configuration

These steps show how to manually configure your client system to register to **Red Hat Satellite**.

1. Make the following changes in **/etc/rhsm/rhsm.conf**:

```
[server]
hostname =[satellite_fqdn]


[rhsm]
baseurl=https://[fqdn_pulp]/pulp/repos/
repo_ca_cert = %(ca_cert_dir)scandlepin-local.pem
ca_cert_dir = /etc/rhsm/ca/
```

2. Change directories to the **ca** directory, remote copy and move the **candlepin-ca.crt** certificate:

```
# cd /etc/rhsm/ca
scp  [satellite.fqdn]:/etc/candlepin/certs/candlepin-ca.crt .
mv candlepin-ca.crt candlepin-local.pem
```

Result

The client system should now be ready to register to the **Red Hat Satellite**.

Report a bug

10.4. Registration

10.4.1. Registering a Host

These steps show how to register hosts in Red Hat Satellite Server. Hosts provisioned by Satellite Server appear on the **All Hosts** page accessible through **Hosts → All hosts**. Hosts registered to the Satellite Server via Red Hat Subscription Manager, which can occur either during the post phase of a kickstart or through the terminal, will appear on the **Content Hosts** page accessible through **Hosts → Content Hosts**.

You must meet the following conditions before continuing with this task:

- Ensure that all steps in Section 10.2, “Configuring Hosts for Registration” have been completed.
- Make sure there is a pre-existing activation key for the system or create an activation key for the system. See Section 7.1, “Creating an Activation Key” for instructions on creating an activation key.

Procedure 10.2. Registering Systems

1. Open a terminal console and login as root on the command line.
2. Clear old system data in preparation for registering. This makes sure that your updated system data is uploaded properly.

```
subscription-manager clean
```

3. Register the system using the Red Hat Subscription Manager (RHSM):

```
# subscription-manager register --org [your_org_name] --
activationkey [your_activation_key]
```

Note

Activation keys will allow you to add environments, provisioning templates and dictate what subscriptions are available and should be applied to the registering system.

There are various options that may be added. For more information, use the command **man subscription-manager**.

Result:

The command line output after the registration should look like:

```
# subscription-manager register --org MyOrg --activationkey TestKey-1
The system has been registered with id: 62edc0f8-855b-4184-b1b8-
72a9dc793b96
```

The system should now appear in the **Red Hat Satellite** Server.

Note

For systems with Red Hat Enterprise Linux 6.3, the release version defaults to version 6.0. To ensure that it is pointing to the 6.3 repository, please follow these steps:

1. On **Red Hat Satellite**, select **Hosts** → **Content Hosts**.
2. Select the system that needs to be changed and click **Edit**.
3. Click on the **Operating System** tab.
4. Select '6.3' from the **Operating system** drop down menu.
5. Click the **Save** button.

Report a bug

10.4.2. Installing the Katello Agent

Prerequisite

The Red Hat Common repository must be enabled in the Red Hat Satellite Server as it provides the required packages.

These steps show how to install and enable the Katello agent. The katello-agent must be enabled so that the Red Hat Satellite Server or Capsule Server can provide information about errata that are applicable for the system.

To install katello-agen:

1. Open a terminal console and login as root on the command line.
2. Install the katello-agent using the following command:

```
# yum install katello-agent
```

Result:

The Katello agent has been installed and enabled. Packages can now be remotely installed and removed from the **Satellite** server.

Report a bug

10.4.3. Installing and Configuring the Puppet Agent

These steps show how to install and configure the Puppet agent on a host. The Puppet agent must be configured so that Red Hat Satellite Server can list all available hosts in **`All Hosts`**.

Procedure 10.3. Installing and Enabling the Puppet Agent

1. Open a terminal console and login as root on the command line.
2. Install the Puppet agent using the following command:

Note

The Red Hat Common repository must be enabled as it provides Puppet.

```
# yum install puppet
```

Result:

The Puppet agent has been installed and enabled.

Procedure 10.4. Configuring the Puppet Agent

Prerequisites

You must meet the following conditions before continuing with this task:

- The host must be registered to the Red Hat Satellite Server.

- The Red Hat Common repository must be enabled.
- Puppet packages must be installed on the host.

1. Configure the Puppet agent by changing the **/etc/puppet/puppet.conf** file:

```
# vi /etc/puppet/puppet.conf
```

```
[main]
    # The Puppet log directory.
    # The default value is '$vardir/log'.
    logdir = /var/log/puppet

    # Where Puppet PID files are kept.
    # The default value is '$vardir/run'.
    rundir = /var/run/puppet

    # Where SSL certificates are kept.
    # The default value is '$confdir/ssl'.
    ssldir = $vardir/ssl

[agent]
    # The file in which puppetd stores a list of the classes
    # associated with the retrieved configuratiion.  Can be loaded
in
    # the separate ``puppet`` executable using the ``--
loadclasses``
    # option.
    # The default value is '$confdir/classes.txt'.
    classfile = $vardir/classes.txt
    pluginsync = true
    report = true
    ignoreschedules = true
    daemon = false
    ca_server = satellite.example.com
    server = satellite.example.com

    # Where puppetd caches the local configuration.  An
    # extension indicating the cache format is added automatically.
    # The default value is '$confdir/localconfig'.
    localconfig = $vardir/localconfig
```

2. Run the Puppet agent on the host:

```
# puppet agent -t --server satellite.example.com
```

3. Sign the SSL certificate for the puppet client through the Satellite Server web interface:

    a. Log into the Satellite Server through the web interface.

    b. Select **Infrastructure** → **Capsules**.

    c. Click **Certificates** to the right of the required host.

    d. Click **Sign**.

Note

Once the Puppet agent is configured on the host it will be listed under **All Hosts** but only when **Any Context** is selected as the host will not be assigned to an organization or location. To assign the host to an organization see Section 3.1.3, “Editing an Organization” and to assign the host to a location see Section 3.2.2, “Editing a Location”.

Result:

The Puppet agent has been configured and now all hosts appear under **All Hosts**.

Report a bug

10.5. Removing a Host

To remove a host from Red Hat Satellite:

1. Click **Hosts** → **All hosts** or **Hosts** → **Content Hosts**.
2. Choose the hosts to be removed.
3. Click **Select Action** and choose **Delete Hosts** from the drop down menu.
4. A confirmation pop-up box will appear. Select **Yes** to remove the host from **Red Hat Satellite** permanently.

Result

The host is removed from **Red Hat Satellite**.

Report a bug

Chapter 11. Using the Foreman Discovery Plugin

The Foreman Discovery plugin in Red Hat Satellite Server adds Metal-as-a-Service (MAAS) features. Bare metal hosts on the Satellite Server managed networks can be booted over the network through PXE into stripped-down Red Hat Enterprise Linux systems running from memory to collect and send hardware facts to the Satellite Server.

Once booted, these systems appear as discovered hosts within the Satellite Server. The hardware facts that are collected via the Discovery plugin allow administrators to provision these systems from these facts, removing the need to manually collect MAC addresses and other hardware information and shortening the time to provision.

Once the provisioning configuration is complete, the Foreman discovery plugin will send the order to the discovered host to reboot and start the installation. The host will then move from the **Discovered Hosts** list in the Satellite Server to the **All Hosts** list.

Prerequisites

Using the Foreman Discovery Plugin requires the following prerequisites:

- A Red Hat Satellite Server with a minimum version of 6.0.5 with bare metal provisioning configured.
- At least one Capsule Server with DHCP and TFTP services enabled.
- At least one host that can be discovered with at least 1GB RAM.

11.1. Installing the Foreman Discovery Plugin

To utilize the Foreman Discovery feature, follow these steps to install the Foreman Discovery Plugin.

1. Open a terminal console and login as root on the command line.
2. Install the Foreman Discovery Plugin using the following command:

```
# yum install foreman-discovery-image
```

Report a bug

11.2. Configuring the Foreman Discovery Plugin

Prerequisites

The following must be configured correctly before the foreman discovery plugin is configured:

- Make sure that bare metal provisioning is functional. See Chapter 9, *Configuring the Provisioning Environment* for more information. To test bare metal provisioning, create a new host entry with a MAC address and power on the system to provision it. Once Foreman Discovery is set up, the host creation will be automated.
- The network DHCP server must be able to serve unknown clients and it must point unknown clients to the TFTP server that the discovery image resides in. If Internet Systems Consortium (ISC) DHCP was installed via the capsule installer on the Red Hat Enterprise Linux host, this will already be configured. If you are using a different configuration or a different DHCP server ensure

the nextserver option is configured to return the proper TFTP server, which is under the Red Hat Satellite Server's control.

These steps show how to configure the Foreman Discovery Plugin:

Procedure 11.1. Configuring the Foreman Discovery Plugin

1. A template needs to be unlocked as it cannot be changed in the user interface. Start a Rails console on the Satellite Server and run the following command:

```
# foreman-rake console
> ct = ConfigTemplate.find_by_name("PXELinux global default");
ct.locked = false; ct.save!
> exit
```

2. Click **Hosts** → **Provisioning templates** and edit the PXELinux global default template. Add the following entry at the end of the template:

```
LABEL discovery
MENU LABEL Foreman Discovery
MENU DEFAULT
KERNEL boot/foremandiscoveryimagelatest.el6.isovmlinuz
APPEND rootflags=loop
initrd=boot/foremandiscoveryimagelatest.el6.isoimg
root=live:/foreman.iso rootfstype=auto ro rd.live.image
rd.live.check rd.lvm=0 rootflags=ro crashkernel=128M
elevator=deadline max_loop=256 rd.luks=0 rd.md=0 rd.dm=0 nomodeset
selinux=0 stateless foreman.url=<%= foreman_url %>
IPAPPEND 2
```

   The **foreman.url** option on the APPEND line defines where the Foreman instance is. Make sure that this is set correctly in global settings or the discovered hosts will not register to Foreman.

   The **IPAPPEND 2** option is important as it adds the **BOOTIF=MAC** option. This is reported via facter as **discovery_bootif**, which is a key fact used for provisioning. It is expected that the interface that it is booted from will be the provisioning interface as well. The same interface will be used as the primary interface and is used to retrieve DNS configuration from DHCP.

3. Make the **foreman.url** option the default by altering the ONTIMEOUT option:

```
ONTIMEOUT discovery
```

4. Alternatively, the discovery image can search for a DNS SRV record named **_xforeman._tcp**. If the DNS server is configured for this, then the **foreman.url** option should not be configured otherwise it will override the DNS SRV record. An example of the configuration for ISC DNS server is:

```
_xforeman._tcp SRV 0 5 443 foreman
```

5. Click **Hosts** → **Provisioning templates** then click **Build PXE Default** to deploy the configuration file on the TFTP server.

Note

The Foreman Discovery image does not support SELinux and the option **`selinux=0`** must be configured in the template. The discovery image is read only.

The Foreman Discovery Plugin is now configured.

Report a bug

11.3. Editing the Discovered Host

An unknown host is not visible in the Red Hat Satellite Server web interface until the uknown host boots, reports, and registers to Foreman. Only then is the host discovered and available on the Satellite Server interface for editing. At the time of reporting, the discovered host will provide Foreman with system details which populates the Discovered Host's profile.

Now that the discovered host is in the Red Hat Satellite Server web interface, edit the Discoved Host's provisioning profile to allow the Satellite Server to provision the host with the correct requirements.

Procedure 11.2. Editing the Discovered Host

1. Click **Hosts** → **Discovered Hosts**.
2. Select the host and click **`Provision`**.
3. Edit the required details.
4. Click **Save**.

Once changes are made to the provisioning details and saved, the discovered host will reboot and install the chosen operating system with the correct requirements.

Report a bug

11.4. Troubleshooting for the Foreman Discovery Plugin

This section provides information about troubleshooting the Foreman Discovery Plugin. For example, instances where the unknown hosts booting up and failing to register with Foreman have a number of common causes.

- If the machine fails to boot to the correct image, verify that the **`/var/lib/tftp/pxelinux.cfg/default`** has been configured as detailed in Section 11.2, “Configuring the Foreman Discovery Plugin”
- If the machine booted the correct image but failed to contact Foreman, check the **`foreman.url`** option on the PXELinux Template. Check that the DNS is working for the image or add an IP address in the **`foreman.url`** option to check if the DHCP is handling IPs to the booted image correctly.
- If the root account is locked on the image and ssh access is disabled but log in access is still available on the terminal. Provide the **`rootpw`** option on the command line. To generate a salted password use:

```
$ openssl passwd salt RH redhat
RHhwCLrQXB8zE
```

Report a bug

Chapter 12. Configuring Host Collections

The *Host Collections* application tab is a system management tool that allows the administrator to:

- Add hosts to a collection.
- Apply a mass installation of packages, errata, or package groups to all host members of a host collection.
- Update specific packages, errata, or specific package groups to all host members.

Report a bug

12.1. Creating a Host Collection

These steps show how to create Host Collections in **Red Hat Satellite**.

Procedure 12.1. Create Host Collections

1. Click **Hosts** → **Host Collections**.
2. Click the **New Host Collection** button.
3. Add the Name and Description of the Host Collection.
4. Uncheck the **Unlimited Content Hosts** button to specify the maximum number of hosts that will be allowed to the group. Otherwise, leave it checked to allow unlimited hosts to join the host collection.
5. Click the **Save** button.

Result:

A new host collection is created.

Report a bug

12.2. Adding Hosts to a Host Collection

These steps show how to add hosts to host collections in **Red Hat Satellite**.

Task Prerequisites

You must meet the following condition before continuing with this task:

- A host must be registered to **Red Hat Satellite** in order to add it to a Host Collection. Refer to Section 10.4.1, “Registering a Host” for information on how to register a host.

Procedure 12.2. Create Host Collections

1. Click **Hosts** → **Host Collections**.
2. Click the host collection where the host should be added.
3. Click the **Content Hosts** subtab.
4. Click the **Add** tab.

5. Check the box next to the desired host and click the **Add Selected** button.

Result:

A host is added to the host collection.

Report a bug

12.3. Adding Content to Host Collections

12.3.1. Adding Packages to a Host Collection

These steps show how to add packages to host collections in **Red Hat Satellite**.

Task Prerequisites

You must meet the following conditions before continuing with this task:

- The content to be added should be available in one of the existing repositories or added prior to this procedure.
- Content should be promoted to the environment where the hosts are assigned.

Procedure 12.3. Adding Packages to Host Collections

1. Click **Hosts** → **Host Collections**.
2. Click the host collection where the package should be added.
3. Click the **Collection Actions** subtab.
4. Click the **Package Installation, Removal, and Updates** link.
5. In the field provided, type in the package or package group name. Then click:
   - **Install** - if you wish to install a new package
   - **Update** - if you wish to update an existing package in the host collection

Result:

The selected packages or package groups within the host collection will be installed or updated.

Report a bug

12.3.2. Adding Errata to a Host Collection

These steps show how to add errata to host collections in **Red Hat Satellite**.

Prerequisites

You must meet the following conditions before continuing with this task:

- The errata to be added should be available in one of the existing repositories or added prior to this procedure.
- Errata should be promoted to the environment where the hosts are assigned.

Procedure 12.4. Adding Errata to Host Collections

1. Click **Hosts** → **Host Collections**.
2. Choose the host collection where the errata should be added.
3. Click the **Collection Actions** subtab.
4. Click the **Errata Installation** link.
5. Choose the errata you wish to push to the host collection and click **Install Selected**.

Result:

The selected errata will be installed in the hosts within the host collection.

Report a bug

12.4. Removing Content from a Host Collection

These steps show how to remove packages from host collections in **Red Hat Satellite**.

Procedure 12.5. Removing Content from Host Collections

1. Click **Hosts** → **Host Collections**.
2. Click the host collection where the package should be removed.
3. Click the **Collection Actions** subtab.
4. Click the **Package Installation, Removal, and Updates** link.
5. In the field provided, type in the package or package group name. Then click **Remove**.

Result:

The package or package group will be removed from all hosts within the host collection.

Report a bug

12.5. Changing the Assigned Lifecycle Environment or Content View for a Host Collection

These steps show change the assigned lifecycle environment or content view of a host collections in **Red Hat Satellite**.

Procedure 12.6. Changing the Assigned Lifecycle Environment or Content View

1. Click **Hosts** → **Host Collection**.
2. Choose the host collection where the lifecycle enviroment or content view should be changed.
3. Click the **Collection Actions** subtab.
4. Select **Change assigned Lifecycle Environment or Content View**.

5. Select a lifecycle environment by checking the checkbox next to the required lifecycle environment.
6. Select the required content view.
7. Click **Assign**.

Result:

The assigned lifecycle environment or content view has been updated.

Report a bug

12.6. Removing a Host from a Host Collection

These steps show how to remove hosts from host collections in **Red Hat Satellite**.

Procedure 12.7. Remove Hosts from Host Collections

1. Click **Hosts** → **Host Collections**.
2. Choose the desired Host Collection.
3. Click the **Content Hosts** subtab.
4. Check the box next to the host you wish to remove from the host collection.
5. Click the **Remove Selected** button to remove the host from the host collection.

Result:

A host is removed from the host collection.

Report a bug

12.7. Removing a Host Collection

These steps show how to remove a host collection in **Red Hat Satellite**.

1. Click **Hosts** → **Host Collections**.
2. Choose the host collection to be removed.
3. Click the **Remove** button. An alert box appears:

```
Are you sure you want to remove host collection Host Collection Name?
```

4. Click the **Remove** button.

Result:

The host collection is removed.

Report a bug

12.8. Cloning a Host Collection

12.8. Cloning a Host Collection

These steps show how to clone a host collection in **Red Hat Satellite**.

1. Click **Hosts** → **Host Collections**.
2. On the left hand panel, click the host collection you wish to clone.
3. On the right hand corner of the host collection details, click **`Copy Collection`**.
4. Add the desired name of the newly cloned host collection.
5. Click the **`Create`** button.

Result

A cloned copy of the host collection is created.

Report a bug

12.9. Reviewing Host Collection Actions

Prerequisites

Requires an existing host collection.

Procedure 12.8. Reviewing Host Collection Actions

1. Click **Hosts** → **Host Collections**.
2. Click the host collection you wish to view the actions of.
3. Click on the **`Details`** subtab.

Result

All events history and actions performed on the host collection is displayed.

Report a bug

Chapter 13. Red Hat Satellite Capsule Servers

The Red Satellite Capsule Server is a Satellite component that provides federated services to discover, provision, and configure hosts outside of the primary Satellite server. A Satellite Capsule Server provides the following features:

- Pulp Server/Content Node features, including:
  - Repository synchronization
  - Content delivery
- Red Hat Satellite Provisioning Smart Proxy features, including:
  - DHCP, including ISC DHCP servers
  - DNS, including Bind and MS DNS servers
  - Any UNIX-based TFTP server
  - Puppet Master servers from 0.24
  - Puppet CA to manage certificate signing and cleaning
  - Baseboard Management Controller (BMC) for power management

The Satellite Capsule Server is a means to scale out the Satellite installation. Organizations can create various capsules in different geographical locations where the data centers are located. These are centrally managed through the Satellite Server. When a Satellite user promotes content to the production environment, the Satellite Server will push the content from the Satellite Server to each of the Satellite Capsule Servers. Host systems pull content and configuration from the Satellite Capsule Servers in their location and not from the central Satellite Server.

Creating various Satellite Capsule Servers will decrease the load on the central server, increase redundancy, and reduce bandwidth usage.

Report a bug

13.1. Red Hat Satellite Capsule Server Scalability

The maximum number of Capsule Servers that the Satellite Server can support has no fixed limit but has been tested on a Satellite Server with a Red Hat Enterprise Linux 6.5 and 7 hostsystems. Currently, running fourteen capsules with two vCPUs have been tested to run without issues.

Capsule scalability depends heavily on the following factors, especially when managing puppet clients:

1. Number of CPUs
2. Run-interval distribution
3. Number of puppet classes

The Capsule Server has a concurrency limitations of 100 concurrent puppet agents running at any single point in time. Running more than 100 concurrent puppet agents will result in a 503 HTTP error.

For example, assuming that the puppet agent runs are evenly distributed with less than 100 concurrent puppet agents running at any single point during a run-interval, a Capsule Server with

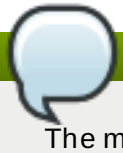
four CPUs can expect a maximum of 1250-1600 puppet clients with a moderate workload of 10 puppet classes assigned to each puppet client. Depending on the number of puppet clients required, the Satellite installation can scale out the number of Capsule Servers to support them.

Based on the following assumptions:

1. There are no external puppet clients reporting directly to the Satellite 6 integrated capsule.
2. All other puppet clients report directly to an external capsule.

Puppet scalability within Satellite on Red Hat Enterprise Linux 6.5 Capsules are as follows:

1. On the minimum amount of CPUs (two CPUs):
    a. At 1 puppet class per host: Not tested
    b. At 10 puppet classes per host: Maximum of 1020-860
    c. At 20 puppet classes per host: Maximum of 375-330
2. On the recommended amount of CPUs (four CPUs):
    a. At 1 puppet class per host: Maximum of 2250-1875
    b. At 10 puppet classes per host: Maximum of 1600-1250
    c. At 20 puppet classes per host: Maximum of 700-560



Note

The maximums in the above given numbers represent an evenly distributed run interval of all puppet agents. Any deviation runs the risk of filling the passenger request queue and is subject to the concurrency limitation of 100 concurrent requests.

Report a bug

13.2. Red Hat Satellite Capsule Server Prerequisites

The Satellite Capsule's requirements are identical to the Satellite Server. These conditions must be met before installing Red Hat Satellite Capsule:

Base Operating System

Red Hat Satellite Capsule is supported on Red Hat Enterprise Linux 6.5 or later, as well as Red Hat Enterprise Linux 7. Install the operating system from disc, local ISO image, kickstart, or any other methods that Red Hat supports. Red Hat Satellite Capsule requires Red Hat Enterprise Linux installations with the @Base package group with no other package-set modifications, and without third-party configurations or software that is not directly necessary for the direct operation of the server. This restriction includes hardening or other non-Red Hat security software. If such software is required in your infrastructure, install and verify a complete working Red Hat Satellite Capsule first, then create a backup of the system before adding any non-Red Hat software.

When installing Red Hat Enterprise Linux from CD or ISO image, there is no need to select any package groups; Red Hat Satellite Capsule only requires the base operating system installation. When installing the operating system via kickstart, select the @Base package group.

- There should be at least one networked host with the following minimum specifications:
  - 64-bit architecture
  - Red Hat Enterprise Linux 6.5 or later
  - A minimum of two CPU cores, but four CPU cores are recommended
  - A minimum of 8 GB memory but ideally 12 GB of memory for each Satellite instance. It is also recommended to use 4 GB of swap space where possible.
  - A minimum of 5 GB storage for the base install of Red Hat Enterprise Linux, 300 MB for the installation of Red Hat Satellite Capsule and at least 10 GB storage for each unique software repository to be synchronized in the /var file system.

    Packages that are duplicated in different channels are only stored once on the disk. Additional repositories containing duplicate packages will require less additional storage.

    Note

    The bulk of storage resides on the **/var/lib/mongodb** and **/var/lib/pulp** directories. These end points are not manually configurable. Ensure that storage is available on the **/var** file system to prevent storage issues.

  - No Java virtual machine installed on the system, remove any if they exist.
  - No *Puppet* RPM files installed on the system
  - No third-party unsupported yum repositories enabled. Third-party repositories may offer conflicting or unsupported package versions that may cause installation or configuration errors.
- Administrative user (**root**) access
- Full forward and reverse DNS resolution using a fully qualified domain name. Check that ***hostname*** and ***localhost*** resolve correctly, using the following commands:

```
# ping -c1 localhost
# ping -c1 `hostname -s` # my_system
# ping -c1 `hostname -f` # my_system.domain.com
```

- Available subscriptions on the Red Hat Satellite Server.

Important

Make sure that the host system is fully updated before installing Red Hat Satellite. Attempts to install on host systems running Red Hat Enterprise Linux that are not fully updated may lead to difficulty in troubleshooting, as well as unpredictable results.

Application Specifications

Satellite application installation specifications are as follows:

It is recommended that a time synchronizer such as **ntpd** is installed and enabled on Satellite. Run the following command to start the **ntpd** service and have it persist across restarts:

```
# service ntpd start; chkconfig ntpd on
```

Required Network Ports

The following conditions must be met before continuing with this task:

- Port 443 for HTTPS (secure WWW) must be open for incoming connections.
- Port 5671 must be open for SSL communication with managed systems.
- Port 80 for HTTP (WWW) must be open to download the bootstrap files.
- Port 8080 for TCP must be free for java connections.
- Port 8140 must be open for incoming Puppet connections with the managed systems.
- Port 9090 must be open for Foreman Smart Proxy connections with the managed systems.

Run the following commands to configure the firewall with the **iptables** command and to make these rules persistent during reboots:

1. For Red Hat Enterprise Linux 6:

```
# iptables -I INPUT -m state --state NEW -p tcp --dport 443 -j
ACCEPT \
&& iptables -I INPUT -m state --state NEW -p tcp --dport 5671 -j
ACCEPT \
&& iptables -I INPUT -m state --state NEW -p tcp --dport 80 -j
ACCEPT \
&& iptables -I INPUT -m state --state NEW -p tcp --dport 8140 -j
ACCEPT \
&& iptables -I INPUT -m state --state NEW -p tcp --dport 9090 -j
ACCEPT \
&& iptables -I INPUT -m state --state NEW -p tcp --dport 8080 -j
ACCEPT \
# iptables-save > /etc/sysconfig/iptables
```

2. For Red Hat Enterprise Linux 7:

```
# firewall-cmd --permanent --add-port="443/tcp" --add-
port="5671/tcp" --add-port="80/tcp" --add-port="8140/tcp" --add-
port="9090/tcp" --add-port="8080/tcp"
# firewall-cmd --complete-reload
```

Red Hat Satellite Server

The Satellite Server must have the *Red Hat Software Collections* repositories enabled and synchronized prior to the Capsule Server installation.

Important

Red Hat recommends that the Satellite Capsule system be a freshly provisioned system that serves no other function except as a Satellite Capsule.

Report a bug

13.3. Installing a Red Hat Satellite Capsule Server

Prerequisites

The Capsule Server must be registered to the Red Hat Satellite Server to use the Red Hat Satellite Server products and subscriptions:

1. Install the Red Hat Satellite Server's CA certificate in the Capsule Server:

```
# rpm -Uvh http://satellite.example.com/pub/katello-ca-consumer-
latest.noarch.rpm
```

2. Register the Capsule Server under your chosen organization's name:

```
# subscription-manager register --org "your organization"
```

This procedure installs a Red Hat Satellite Capsule Server onto a host.

Procedure 13.1. To Install a Satellite Capsule Server on a Certificate-managed System:

1. List all the available subscriptions to find the correct Red Hat Satellite and Red Hat Enterprise Linux product to allocate to your system:

```
 # subscription-manager list --available --all
```

The screen displays:

```
+-------------------------------------------+
    Available Subscriptions
+-------------------------------------------+


ProductName:        Red Hat Satellite
ProductId:          SKU123456
PoolId:             e1730d1f4eaa448397bfd30c8c7f3d334bd8b
Quantity:           10
Multi-Entitlement:  No
Expires:            08/20/2013
MachineType:        physical
```

Note

The SKU and Pool ID depend on the Red Hat Satellite product type that corresponds to your system version and product type.

2. Subscribe to the pool using the following command:

```
# subscription-manager subscribe --pool=Red_Hat_Satellite_Pool_Id
# subscription-manager subscribe --
pool=Red_Hat_Enterprise_Linux_Pool_Id
# subscription-manager subscribe --
pool=Red_Hat_Enterprise_Linux_Software_Collections_Pool_Id
```

3. Disable all existing repositories:

```
# subscription-manager repos --disable "*"
```

4. Enable the Satellite and Red Hat Enterprise Linux repositories by running **subscription-manager**. You might need to alter the Red Hat Enterprise Linux repository to match the specific version you are using.

```
# subscription-manager repos --enable rhel-6-server-rpms \
--enable rhel-server-rhscl-6-rpms \
--enable rhel-6-server-satellite-capsule-6.0-rpms
```

5. Install the *katello-installer* and *cyrus-sasl-plain* packages using the **yum install** command as the root user:

```
# yum install katello-installer cyrus-sasl-plain
```

The *katello-installer* provides the **capsule-installer** functionality while *cyrus-sasl-plain* is required for **pulp**.

Result:

The Satellite Capsule Server is installed on your host system. The Satellite Capsule Server must be configured before it can be used.

Report a bug

13.4. Configuring a Red Hat Satellite Capsule Server

Prerequisite

You must meet the following conditions before continuing on this task:

- Install the Red Hat Satellite Server.
- Set the SELinux permissions on the system designated as the Satellite Capsule Server as enforcing.

The following procedures configure a Satellite Capsule Server for use with your Red Hat Satellite Server. This includes the following types of Satellite Capsule Servers:

- Satellite Capsule Server with Smart Proxy
- Satellite Capsule Server as a Content Node
- Satellite Capsule Server as a Content Node with Smart Proxy

To configure a Satellite Capsule Server:

1. On the Satellite Server:

    a. Generate a Satellite Capsule Server certificate:

    ```
    capsule-certs-generate --capsule-fqdn capsule_FQDN --certs-
    tar ~/capsule.example.com-certs.tar
    ```

    Where:

    - **capsule_FQDN** is the Satellite Capsule Server's fully qualified domain name. (REQUIRED)
    - **certs-tar** is the name of the tar file to be generated that contains the certificate to be used by the Satellite Capsule installer.

    Running **capsule-certs-generate** will generate the following output message:

    ```
        To finish the installation, follow these steps:
      1. Ensure that the capsule-installer is available on the
    system.
         The capsule-installer comes from the katello-installer
    package and
         should be acquired through the means that are
    appropriate to your deployment.
      2. Copy ~/capsule.example.com-certs.tar to the capsule
    system capsule.example.com
      3. Run the following commands on the capsule (possibly with
    the customized
         parameters, see capsule-installer --help and
         documentation for more info on setting up additional
    services):
      rpm -Uvh http://master.com/pub/katello-ca-consumer-
    latest.noarch.rpm
      subscription-manager register --org "Default Organization"
      capsule-installer --parent-fqdn
    "sat6.example.com"\
                        --register-in-foreman  "true"\
                        --foreman-oauth-key
    "xmmQCGYdkoCRcbviGfuPdX7ZiCsdExf
                        --foreman-oauth-secret
    "w5ZDpyPJ24eSBNo53AFybcnqoDYXgLUA"\
                        --pulp-oauth-secret
    "doajBEXqNcANy93ZbciFyysWaiwt6BWU"\
                        --certs-tar
    "~/capsule.example.com-certs.tar"\
                        --puppet               "true"\
                        --puppetca             "true"\
                        --pulp                 "true"
    ```

b. Copy the generated tarball, capsule.example.com-certs.tar, from the Satellite Server to the Satellite Capsule host system.

2. On the Satellite Capsule Server:

a. Register your Satellite Capsule Server to the Satellite Server:

```
# rpm -Uvh http://sat6host.example.redhat.com/pub/katello-ca-
consumer-latest.noarch.rpm
# subscription-manager register --org "Default Organization"
--env [environment]/[content_view_name]
```

Note

The Satellite Capsule Server must be assigned to an organization as the Satellite Capsule Server requires an environment to synchronize content from the Satellite Server. Only organizations have environments.

Assigning a location is optional though recommended to indicate proximity to the hosts that the Satellite Capsule Server is managing.

b. Depending on the desired Satellite Capsule Server type, choose one of the following options:

i. Option 1: Satellite Capsule Server with Smart Proxy: This installs a Satellite Capsule Server with Smart Proxy features (DHCP, DNS, Puppet). Run the following commands as the root user on the Satellite Capsule Server:

```
# capsule-installer --parent-fqdn
"satellite.example.com"\
                    --register-in-foreman  "true"\
                    --foreman-oauth-key
"xmmQCGYdkoCRcbviGfuPdX7ZiCsdExf
                    --foreman-oauth-secret
"w5ZDpyPJ24eSBNo53AFybcnqoDYXgLUA"\
                    --pulp-oauth-secret
"doajBEXqNcANy93ZbciFyysWaiwt6BWU"\
                    --certs-tar
"/root/capsule.example.com-certs.tar"\
                    --puppet                "true"\
                    --puppetca              "true"\
                    --pulp                  "true"
                    --tftp                  "true"
                    --dhcp                  "true"\
                    --dhcp-interface        "virbr1
                   --dns                   "true"\
                   --dns-forwarders        "8.8.8.8"\
                   --dns-forwarders        "8.8.4.4"\
                   --dns-interface         "virbr1"\
                   --dns-zone              "example.com"
```

ii. Option 2 - Satellite Capsule Server as a Content Node with Smart Proxy: This installs a Satellite Capsule Server with all the features. Run the following commands as the root user on the Satellite Capsule Server:

```
# capsule-installer --parent-fqdn
"sat6.example.com"\
                    --register-in-foreman  "true"\
                    --foreman-oauth-key
"xmmQCGYdkoCRcbviGfuPdX7ZiCsdExf
                    --foreman-oauth-secret
"w5ZDpyPJ24eSBNo53AFybcnqoDYXgLUA"\
                    --pulp-oauth-secret
"doajBEXqNcANy93ZbciFyysWaiwt6BWU"\
                    --certs-tar
"/root/capsule.example.com-certs.tar"\
                    --puppet                "true"\
                    --puppetca              "true"\
                    --pulp                  "true"
                    --tftp                  "true"
                    --dhcp                  "true"\
                    --dhcp-interface        "virbr1
                   --dns                   "true"\
                   --dns-forwarders        "8.8.8.8"\
                   --dns-forwarders        "8.8.4.4"\
                   --dns-interface         "virbr1"\
                   --dns-zone              "example.com"
```

3. Run the following commands to configure the firewall to limit **elasticsearch** to the **foreman**, **katello** and root users and make these rules persistent during reboots:

   - For Red Hat Enterprise Linux 6:

```
iptables -A OUTPUT -o lo -p tcp -m tcp --dport 9200 -m owner --
uid-owner foreman -j ACCEPT \
&& iptables -A OUTPUT -o lo -p tcp -m tcp --dport 9200 -m owner
--uid-owner katello -j ACCEPT \
&& iptables -A OUTPUT -o lo -p tcp -m tcp --dport 9200 -m owner
--uid-owner root -j ACCEPT \
&& iptables -A OUTPUT -o lo -p tcp -m tcp --dport 9200 -j DROP
iptables-save > /etc/sysconfig/iptables
```

   - For Red Hat Enterprise Linux 7:

```
firewall-cmd --permanent --direct --add-rule ipv4 filter OUTPUT
0 -o lo -p tcp -m tcp --dport 9200 -m owner --uid-owner foreman
-j ACCEPT \
&& firewall-cmd --permanent --direct --add-rule ipv6 filter
OUTPUT 0 -o lo -p tcp -m tcp --dport 9200 -m owner --uid-owner
foreman -j ACCEPT \
&& firewall-cmd --permanent --direct --add-rule ipv4 filter
OUTPUT 0 -o lo -p tcp -m tcp --dport 9200 -m owner --uid-owner
katello -j ACCEPT \
&& firewall-cmd --permanent --direct --add-rule ipv6 filter
OUTPUT 0 -o lo -p tcp -m tcp --dport 9200 -m owner --uid-owner
```

```
katello -j ACCEPT \
&& firewall-cmd --permanent --direct --add-rule ipv4 filter
OUTPUT 0 -o lo -p tcp -m tcp --dport 9200 -m owner --uid-owner
root -j ACCEPT \
&& firewall-cmd --permanent --direct --add-rule ipv6 filter
OUTPUT 0 -o lo -p tcp -m tcp --dport 9200 -m owner --uid-owner
root -j ACCEPT \
&& firewall-cmd --permanent --direct --add-rule ipv4 filter
OUTPUT 1 -o lo -p tcp -m tcp --dport 9200 -j DROP \
&& firewall-cmd --permanent --direct --add-rule ipv6 filter
OUTPUT 1 -o lo -p tcp -m tcp --dport 9200 -j DROP
```

Note

If the configuration is successful, run this command as the root user on the Satellite Capsule Server:

```
# echo $?
```

This command should return a "0" to indicate success. If it does not, check **/var/log/kafo** to debug the cause of failure. **/var/log/kafo** is the log file for the output generated by the commands **capsule-certs-generate** and **capsule-installer**.

The Satellite Capsule Server should also appear in the Satellite Server's User Interface under **Infrastructure → Capsules**.

Result:

The Satellite Capsule Server is now configured and registered with the Satellite Server.

Report a bug

13.5. Adding Lifecycle Environments to a Red Hat Satellite Capsule Server

If the newly created Red Hat Satellite Capsule Server has *Content Node* features enabled, the Satellite Capsule Server needs an environment added to the Satellite Capsule Server. Adding an environment to the Red Hat Satellite Capsule Server will allow the Satellite Capsule Server to synchronize content from the Satellite Server and provide content to host systems.

Important

The Satellite Capsule Server is configured through the Satellite Server's command line interface (CLI). Execute all **hammer** commands on the Satellite Server.

To add environments to your Satellite Capsule Server:

1. Log in to the Satellite Server CLI as root.
2. Choose the desired Red Hat Satellite Capsule Server from the list and take note of its **id**:

```
# hammer capsule list
```

The Satellite Capsule Server's details can be verified using the command:

```
# hammer capsule info --id capsule_id_number
```

3. Verify the list of lifecycle environments available for the Red Hat Capsule Server and note down the **`environment id`**:

```
# hammer capsule content available-lifecycle-environments --id
capsule_id_number
```

Where:

- **`available-lifecycle-environments`** are lifecycle environments that are available to the Satellite Capsule but are currently not attached to the Satellite Capsule.

4. Add the lifecycle environment to the Satellite Capsule Server:

```
# hammer capsule content add-lifecycle-environment --id
capsule_id_number --lifecycle-environment-id environment_id_number
```

Where:

- **`--id`** is the Satellite Capsule Server's identification number.
- **`--lifecycle-environment-id`** is the lifecycle environment's identification number.

Repeat this step for every lifecycle environment to be added to the Capsule Server.

5. Synchronize the content from the Satellite Server's environment to the Satellite Capsule Server:

```
# hammer capsule content synchronize --id capsule_id_number
```

When a Satellite Capsule Server has various lifecycle environments, and only one lifecycle environment needs to be synchronized, it is possible to target a specific environment by specifying the environment identification:

```
# hammer capsule content synchronize --id 1 --environment-id 1
```

Result:

The chosen environments now consume packages from repositories on the desired Satellite Capsule Server.

Report a bug

13.6. Removing Lifecycle Environments from the Red Hat Satellite Capsule Server

There are multiple reasons to remove lifecycle environments from the Red Hat Satellite Capsule Server. For example:

- When lifecycle environments are no longer relevant to the host systems
- When lifecycle environments have been incorrectly added to the Satellite Capsule Server

To remove a lifecycle environment from the Satellite Capsule Server:

1. Log in to the Satellite Server CLI as the root user.
2. Choose the desired Red Hat Satellite Capsule Server from the list and take note of its **`id`**:

   ```
   # hammer capsule list
   ```

   The Satellite Capsule Server's details can be verified using the command:

   ```
   # hammer capsule info --id capsule_id_number
   ```

3. Verify the list of lifecycle environments currently attached to the Red Hat Capsule Server and note down the **`environment id`**:

   ```
   hammer capsule content lifecycle-environments --id
   capsule_id_number
   ```

4. Remove the lifecycle environment from the Satellite Capsule Server:

   ```
   # hammer capsule content remove-lifecycle-environment --id
   capsule_id_number --environment-id environment_id
   ```

   Where:

   - **`--id`** is the Satellite Capsule Server's identification number.
   - **`--environment-id`** is the lifecycle environment's identification number.

   Repeat this step for every lifecycle environment to be removed from the Capsule Server.

5. Synchronize the content from the Satellite Server's environment to the Satellite Capsule Server:

   ```
   # hammer capsule content synchronize --id capsule_id_number
   ```

Result:

The chosen environments are removed from the Satellite Capsule Server.

Report a bug

13.7. Registering Host Systems to a Red Hat Satellite Capsule Server

Prerequisite

Client system must be configured for registration. The following chapters in the *Red Hat Satellite User Guide* can assist with configuration:

1. *Configuring Hosts for Registration*
2. *Automated Configuration*

3. *Manual Configuration*

4. *Creating a New Activation Key*

Systems can be registered to the parent host while using a Satellite Capsule as a content source. Register the system to the Satellite 6 server through **`subscription-manager`** but reference the Satellite Capsule by using the **`--baseurl`** flag provided by Subscription Manager using /pulp/repos as the prefix.

```
# subscription-manager register --org [org_name] --activationkey
[your_activation_key] --baseurl
https://satcapsule.example.com/pulp/repos
```

Report a bug

13.8. Refreshing a Red Hat Satellite Capsule Server

Procedure 13.2. Refreshing a Red Hat Satellite Capsule Server

1. Select **Infrastructure** → **Capsules**.

2. Select **Refresh Features** from the drop down menu to the right of the name of the capsule you want to refresh.

Result

The capsule is refreshed with the changes that have been made.

Report a bug

Chapter 14. Users and Roles

14.1. Configuring LDAP Authentication for Red Hat Satellite

Red Hat Satellite includes the option to use a Lightweight Directory Access Protocol (LDAP) service for user information and authentication, using one or more LDAP directories.

Procedure 14.1. To Configure LDAP Authentication:

1. Log in as the Satellite administrator.
2. Click **Administer** → **LDAP Authentication** on the upper right of the page.
3. Click **New LDAP source** on the upper right of the page.
4. Fill in the information in the following tabs:
    a. **LDAP Server** Tab. For the LDAP server tab, enter the following information:
        - Name - LDAP server's name.
        - Server - the LDAP server's hostname.
        - Port - the LDAP port. This defaults to port 389.
        - TLS - Enables Transport Layer Security. Tick the check box to enable encryption.

    b. **Account** Tab. For the Account Tab, enter the following information:
        - Account Username - an LDAP user who has read access to the LDAP server. This field cannot remain blank. Use the full path to the user's object. For example:

          ```
          uid=$login,cn=users,cn=accounts,dc=example,dc=com
          ```

        - Account password - the LDAP password for the user defined in the Account username field. This field can remain blank if the Account username is using the "$login" variable.
        - Base DN - the top level domain name of your LDAP directory. For example:

          ```
          cn=users,cn=accounts,dc=redhat,dc=com
          ```

        - LDAP filter
        - Automatically create accounts in Foreman - creates Satellite accounts automatically for LDAP users who log in for the first time in Satellite. Tick the check box to enable this feature.

    c. **Attribute mappings** tab. LDAP attributes such as login, name, and email addresses, need to be mapped to Satellite attributes for LDAP to work on Satellite. The following attributes can be mapped to Satellite:
        - Login name attribute
        - First Name attribute
        - Surname attribute

- Email address attribute
- Photo attribute

5. Click **Submit**.

Result:

LDAP authentication is now configured on the Satellite.

Report a bug

14.2. Users

A *User* defines a set of details for individuals using the system. Users can be assigned default organizations and environments, so that when they create new entities, these defaults are automatically used. Users can also be assigned to a default *role*, which defines the permissions and level of access they have. The roles to which users are assigned, give them rights to see and manage organizations and environments.

Report a bug

14.2.1. Creating a User

These steps show how to create a user.

Procedure 14.2. To Create a User:

1. Click **Administer** → **Users** and then click **New User**.
2. Enter the required details on the **User** tab.
3. On the **Locations** tab, select the required locations for this user.
4. On the **Organizations** tab, select the required organizations for this user.
5. On the **Roles** tab, select the required roles for this user. Active roles are displayed in the right panel.
6. Click **Submit** to create the user.

Report a bug

14.2.2. Editing a User

These steps show how to edit details of an existing user.

Procedure 14.3. Edit an Existing User

1. Click **Administer** → **Users** at the top right corner of the web page.
2. Click the username of the required user. General information about the user will appear on the right.
3. You can modify the user's username, first name, surname, email address, default location, default organization, language, and password in the **User** tab.

4. You can modify the user's assigned locations in the **Locations** tab.
5. You can modify the user's assigned organizations in the **Organizations** tab.
6. You can modify the user's assigned roles in the **Roles** tab.
7. Click the **Save** button to save your changes.

Result:

An existing user is edited.

Report a bug

14.2.3. Removing a User

These steps show how to remove an existing user.

Procedure 14.4. Remove an Existing User

1. Click **Administer** → **Users** at the top right corner of the webpage.
2. Click the **Delete** link to the right of the username you want to delete from the list of available users.
3. An alert box appears:

   ```
   Delete User Name?
   ```

4. Click the **OK** button.

Result:

An existing user is removed.

Report a bug

14.3. Roles

Roles define a set of permissions and access levels. Once a role has been created, users can be assigned that role. In this way, large groups of users can easily be given the same set of permissions.

Report a bug

14.3.1. Creating a Role

These steps show how to create a role.

Procedure 14.5. Create a Role

1. Click **Administer** → **Roles** at the top right corner of the webpage.
2. Click the **New Role** link.
3. Provide a **Name** for the role.

4. Click the **Submit** button to save your new role.

Result:

A role is created.

Report a bug

14.3.2. Adding Permissions to an Existing Role

These steps show how to add permissions of an existing role.

Procedure 14.6. Add Permissions to an Existing Role

1. Click **Administer** → **Roles** at the top right corner of the web page.
2. Select **Add Permission** from the drop down menu to the right of the required role.
3. Select the **Resource type** from the drop down menu.
4. Select the required permissions from the **Permission** list by clicking the required permission to move it to the **Selected Items** list.
5. Select whether the permission is **Unlimited** by clicking the check box.
6. Click the **Next** button. A list of available permissions is listed.
7. Click the **Submit** button to save changes.

Result:

Permissions are added to an existing role.

Report a bug

14.3.3. Removing a Role

These steps show how to remove an existing role.

Procedure 14.7. Remove an Existing Role

1. Click **Administer** → **Roles** at the top right corner of the web page.
2. Select **Delete** from the drop down box to the right of the role to be deleted.
3. An alert box appears:

```
Delete Role?
```

4. Click the **OK** button.

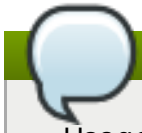
Result:

An existing role is removed.

Report a bug

Chapter 15. Command Line Reference

15.1. Using the Red Hat Satellite Command-Line Interface



Note

Usage of the command-line interface (CLI) is optional.

hammer is the CLI management tool for Red Hat Satellite functionality. It can:

- Provision hosts.
- Edit the attributes of a resource or group.
- Interact and manipulate hosts, capsules and domains.

hammer can be executed on the command line through its parameters and options or through the interactive shell. To invoke the shell:

```
[root@sat.example.com ~]# hammer shell
Welcome to the hammer interactive shell
Type 'help' for usage information
Command completion is disabled on ruby < 1.9 due to compatibility
problems.
hammer> organization list
---|------------------|------------------|------------------------------
ID | NAME             | LABEL            | DESCRIPTION
---|------------------|------------------|------------------------------
1  | ACME_Corporation | ACME_Corporation | ACME_Corporation Organization
3  | Test Corp        | Test_Corp        |
---|------------------|------------------|------------------------------
hammer>
```

The full list of options and subcommands are available on the helpfile:

```
# hammer -h
```

Report a bug

Chapter 16. Maintenance

16.1. Logging and Reporting

Red Hat Satellite provides system information in the form of notifications and logs.

Examples of helpful logs for troubleshooting are:

Table 16.1. Relevant Log Files

Log File	Description
`/var/log/elasticsearch`	Errors concerning the UI search index display
`/var/log/candlepin`	Errors concerning subscription management
`/var/log/foreman`	Errors concerning foreman
`/var/log/foreman-proxy`	Errors concerning the foreman proxy
`/var/log/httpd`	Errors concerning the apache http server
`/var/log/katello-installer`	Errors concerning the Satellite installer
`/var/log/libvirt`	Errors concerning the virtualization API
`/var/log/mongodb`	Errors concerning the database
`/var/log/pulp`	Errors in repository management
`/var/log/puppet`	Errors in configuration management
`/var/log/rhsm`	Errors in the subscription management tool
`/var/log/tomcat6`	Issues concerning the apache webserver

Reports can also be generated to view and monitor information about the hosts being maintained. The **`foreman-debug`** command collects configuration and log data for Red Hat Satellite, its back-end services and system information. This information is collected into a tarball.

Important

`foreman-debug` removes all security information such as password, tokens and keys while collecting information. However, the tarball can still contain sensitive information about the Red Hat Satellite Server. It is recommended to send this information directly to the intended recipient and not publicly.

Report a bug

16.1.1. Viewing Import History

These steps show how to view an import history in **Red Hat Satellite**.

Procedure 16.1. Viewing Import History

1. Click **Content** → **Red Hat Subscriptions**.
2. Click the **`Manage Manifest`** button.
3. Click the **`Import History`** tab.

Result:

Details of the import history are displayed.

Report a bug

16.2. Disaster Recovery

16.2.1. Red Hat Satellite Backup Procedure

Procedure 16.2. Red Hat Satellite Backup Procedure

This example uses the **/backup** directory as our target directory to hold backup archives. Execute the following commands as the **root** system account.

Note

When SELinux is enabled users need to ensure that local content they want to synchronize is labeled with "httpd_sys_content_t".

1. Prepare your backup location:

```
umask 0027
export BDIR=/backup
mkdir $BDIR
chgrp postgres $BDIR
chmod 770 $BDIR
cd $BDIR
```

2. Backup the configuration and data files:

```
# tar --selinux -czvf config_files.tar.gz \
/etc/katello \
/etc/elasticsearch \
/etc/candlepin \
/etc/pulp \
/etc/gofer \
/etc/pki/katello \
/etc/pki/pulp \
/etc/qpidd.conf \
/etc/sysconfig/katello \
/etc/sysconfig/elasticsearch \
/root/ssl-build \
/var/www/html/pub/* \

# tar --selinux -czvf elastic_data.tar.gz /var/lib/elasticsearch
```

3. Back up the repositories. The RPM files in repositories already use compression and cannot be compressed any further. Therefore, depending on instance size, the resulting backup archive (**pulp_data.tar**) can grow quite large.

   There are two options for backing up repositories.

   #### Option One: Online Repositories Backup

Perform a checksum of all timestamps, backup the repository and perform the checksum again.

```
# find /var/lib/pulp -printf '%T@\n' | md5sum
# tar --selinux -cvf pulp_data.tar /var/lib/pulp /var/www/pub
# find /var/lib/pulp -printf '%T@\n' | md5sum
```

If both checksums match, the online backup is correct and usable. If the checksums do not match, perform the repository backup again.

Note

Use the **rsync** command to speed up file copying so checksums match.

Option Two: Offline Repositories Backup

Stop the Pulp server and perform the backup. After the backup has completed, restart the Pulp server.

```
# service pulp-server stop
# tar --selinux -cvf pulp_data.tar /var/lib/pulp /var/www/pub
# service pulp-server start
```

Note

When the Pulp server is down, Red Hat Satellite and the yum clients will be unable to maintain the connection. Any repository actions performed on Red Hat Satellite will fail during this inactivity.

4. Back up your databases:

Offline Database Backup

Perform the offline backup by using these commands:

```
tar --selinux -czvf mongo_data.tar.gz /var/lib/mongodb
tar --selinux -czvf pgsql_data.tar.gz /var/lib/pgsql/data/
```

Warning

Red Hat Satellite must be completely inactive to do an offline backup. Performing this step while Satellite is running may result in corrupted data.

This method archives all data from the PostgreSQL and MongoDB databases. It is recommended to perform this backup during maintenance periods.

PostgreSQL Databases Backup

Check the names for Red Hat Satellite PostgreSQL databases. Default database names are **foremanschema** and **candlepin**. Find all default values in the **katello-configure** manual page, or in the **/usr/share/katello/install/default-answer-file**. If the Red Hat Satellite instance is not using the default names, check the **katello-configure.conf**, which stores all options provided to the **katello-configure** command, and look for **db_name**.

```
# grep db_name /etc/katello/katello-configure.conf
```

Run the **pg_dump** command to create online database backups. It is not necessary to stop PostgreSQL or Red Hat Satellite, and this process does not block logged users. However, the process can take several minutes to finish depending on database sizes.

```
# runuser - postgres -c "pg_dump -Fc foremanschema >
/backup/foreman.dump"
# runuser - postgres -c "pg_dump -Fc candlepin >
/backup/candlepin.dump"
```

Note

For more information about PostgreSQL backups, consult the **pg_dump** manual page or visit http://www.postgresql.org/docs/8.4/static/backup.html.

MongoDB Database Backup

Use online tools to backup the pulp database (MongoDB) while the database is running. Execute the following command in the backup directory:

```
# mongodump --host localhost --out $BDIR/mongo_dump
```

This tool creates the **/backup/mongo_dump/pulp_database** directory, including JSON files.

Note

For more information, consult the **mongodump** man page or visit http://www.mongodb.org/display/DOCS/Backups.

5. Check the backup process. The process creates the following archive files and a directory:

```
# ls $BDIR
candlepin.dump
config_files.tar.gz
elastic_data.tar.gz
katello.dump
mongo_dump/
pulp_data.tar
```

If you performed the optional offline backup of the databases, the following files will also appear:

```
mongo_data.tar.gz
pgsql_data.tar.gz
```

Red Hat Satellite is now backed up. Red Hat Satellite should be online.

Report a bug

16.2.2. Red Hat Satellite Restore Procedure

Procedure 16.3. Red Hat Satellite Restore Procedure

Important

The following process describes a full **Red Hat Satellite** restore. This process deletes all data from a current **Red Hat Satellite** instance. Please make sure that:

- You are restoring to the correct instance. The **Red Hat Satellite** instance must have the same configuration, package versions and errata as the original system.
- All commands are executed as **root** in the directory with archives created during the backup process.
- All SELinux context are correct. Run the command:

```
restorecon -Rnv /
```

1. Prepare the **Red Hat Satellite** host for restore. This process restores the backup on the same server that generated the backup. If the original system is unavailable, provision the same configuration with the same settings (hostname, IP address, etc) and run **katello-installer** with the same options using the file from the configuration backup:

```
# tar --selinux -xzvf config_files.tar.gz -C /tmp
# katello-installer --answer-file=/etc/foreman/database.yml
```

2. Change to the backup directory.

```
# export BDIR=/backup
# chgrp postgres -R $BDIR
# cd $BDIR
```

3. Check the names for **Red Hat Satellite** PostgreSQL databases. Default database names are **foreman** and **candlepin**. Find all default values in the **/etc/foreman/database.yml**. If the **Red Hat Satellite** instance is not using the default names, check the **katello-configure.conf**, which stores all options provided to the **katello-configure** command, and look for **db_name**.

```
# grep db_name /etc/katello/katello-configure.conf
```

4. Stop all services prior to restoring the databases:

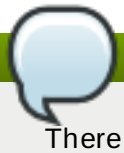
```
# katello-service stop
# service postgresql stop
```

5. Restore the system files. Make sure that the files extract on the correct host. Restore all system files with the following commands executed from the backup directory:

```
# tar --selinux -xzvf config_files.tar.gz -C /
# tar --selinux -xzvf elastic_data.tar.gz -C /
# tar --selinux -xvf pulp_data.tar -C /
```

6. Drop the existing **Red Hat Satellite** PostgreSQL databases if any exist:

```
# service postgresql start
# runuser - postgres -c "dropdb foremanschema"
# runuser - postgres -c "dropdb candlepin"
```



Note

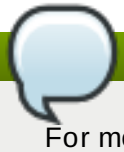
There is a possibility of the following error appearing during database drop:

```
database xxx is being accessed by other users
```

Ensure all processes are stopped.

7. Restore **Red Hat Satellite** PostgreSQL databases with the following commands:

```
# runuser - postgres -c "pg_restore -C -d postgres
/backup/katello.dump"
# runuser - postgres -c "pg_restore -C -d postgres
/backup/candlepin.dump"
```



Note

For more info about PostgreSQL backups consult the **pg_dump** manual page or visit http://www.postgresql.org/docs/8.4/static/backup.html.

8. Ensure MongoDB is running and delete the old data:

```
# service mongod start
# echo 'db.dropDatabase();' | mongo pulp_database
```

9. Execute the following command while providing the **--username** and **--password** in the directory of backup archives.

```
# mongorestore --host localhost mongo_dump/pulp_database/
```

Note

For more information consult **mongorestore** man page or visit http://www.mongodb.org/display/DOCS/Backups.

10. Restart all **Red Hat Satellite** processes:

```
# service postgresql start
# katello-service start
```

11. Check log files for errors, check **audit.log** for denials and attempt to ping the **Red Hat Satellite** instance:

```
# hammer -u admin -p admin ping
```

Report a bug

16.3. Troubleshooting

16.3.1. Changing Your Red Hat Satellite's Fully Qualified Domain Name (FQDN)

Prerequisite

The Satellite FQDN has been changed correctly and the **/etc/sysconfig/network** in Red Hat Enterprise Linux 6 or the **/etc/hostname** file in Red Hat Enterprise Linux 7 has been modified accordingly.

Upon installation, the host system's FQDN is recorded by the Red Hat Satellite Server. Changing the FQDN without the proper precautions can prevent the Satellite Server from running correctly. It also renders all custom server certificates incorrect.

Procedure 16.4. Updating Your Red Hat Satellite Configuration After an FQDN Change

To make sure that Red Hat Satellite continues to run properly even with the FQDN change, follow these steps:

1. Verify that the FQDN is being properly repored and reflects the hostname/FQDN:

```
# facter fqdn
```

2. Update the katello-installer answer file to replace the old FQDN with the new one:

```
# sed -i "s/$OLD_FQDN/$NEW_FQDN/g" /etc/katello-
installer/answers.katello-installer.yaml
```

Where:

a. $OLD_is the Satellite Server's previous FQDN.

b. $NEW_FQDN is the Satellite Server's new FDQN.

3. Delete the amqp-client certificate from the NSS database:

```
# certutil -D -d '/etc/pki/katello/nssdb' -n 'amqp-client'
```

4. Regenerate the server certificates by running **katello-installer**:

```
# katello-installer --certs-update-all
```

5. On client systems registered to the Red Hat Satellite Server, uninstall the existing katello-ca-consumer package since it contains the existing SSL certificate with the old FQDN information and update the package from the new FQDN:

```
# rpm -e $(rpm -qa "katello-ca-consumer*")
#rpm -Uvh http://NEW_FQDN/pub/katello-ca-consumer-latest.noarch.rpm
```

The Red Hat Satellite is now fully updated with the new FQDN and server certificates.

Chapter 17. Configuring Identity Management in Red Hat Satellite

Identity Management (IDM) deals with the management of individual identities, their credentials and privileges used in a networking environment. IDM can help to increase the security of your system and ensure that the right people have access to the right information when they need it.

Red Hat Satellite has a realm feature that will automatically manage the lifecycle of any system registered to a realm or domain provider. This section will explain how you need to configure the Satellite Server or Capsule Server for IDM and how to automatically add client systems to the Satellite 6 Identity Management host group.

Report a bug

17.1. Configuring Red Hat Satellite Server or Capsule Server for IDM Realm Support

The initial step to use Identity Management (IDM) in Red Hat Satellite is to configure the Red Hat Satellite Server or Red Hat Satellite Capsule Server.

Prerequisites

Make sure that the following are setup before configuring IDM:

1. A Satellite Server registered to the content delivery network or an independent Capsule Server registered to the Satellite Server
2. A realm or domain provider such as Red Hat Identity Management configured and set up

To configure the Satellite Server or Capsule Server for IDM Realm Support:

1. On the Satellite Server or Capsule Server, install the following packages:

```
# yum install ipa-client foreman-proxy ipa-admintools
```

2. Configure the Satellite Server (or Capsule Server) as an IPA client:

```
# ipa-client-install
```

3. Create a realm-capsule user and the relevant roles in Red Hat Identity Management on the Satellite Server or Capsule Server:

```
# foreman-prepare-realm admin realm-capsule
```

   Running foreman-prepare-realm will prepare a FreeIPA or Red Hat Identity Management server for use with the Foreman Smart Proxy. It creates a dedicated role with the permissions needed for Foreman, creates a user with that role and retrieves the keytab file. You will need your Identity Management server configuration details on this step.

   If the command successfully executes, you should be able to see the following command output:

```
Keytab successfully retrieved and stored in: freeipa.keytab
Realm Proxy User:    realm-capsule
Realm Proxy Keytab:  /root/freeipa.keytab
```

4. Move the **/root/freeipa.keytab** to the **/etc/foreman-proxy** directory and set the ownership settings to the user foreman-proxy:

```
# mv /root/freeipa.keytab /etc/foreman-proxy
# chown foreman-proxy:foreman-proxy /etc/foreman-
proxy/freeipa.keytab
```

5. Configure the realm based on whether you are using Satellite Server or Capsule Server:

   A. If you are using the integrated capsule in the Satellite Server, use **katello-installer** to configure the realm:

```
# katello-installer --capsule-realm true \
  --capsule-realm-keytab /etc/foreman-proxy/freeipa.keytab \
  --capsule-realm-principal 'realm-capsule@EXAMPLE.COM' \
  --capsule-realm-provider freeipa
```

Note

These options may also be run at the initial configuration of Red Hat Satellite Server.

   B. If you are using an independent Capsule Server, use **capsule-installer** to configure the realm:

```
# capsule-installer --realm true \
  --realm-keytab /etc/foreman-proxy/freeipa.keytab \
  --realm-principal 'realm-capsule@EXAMPLE.COM' \
  --realm-provider freeipa
```

6. Make sure that the most updated versions of the ca-certificates package is installed and trust the IPA Certificate Authority:

```
# cp /etc/ipa/ca.crt /etc/pki/ca-trust/source/anchors/ipa.crt
# update-ca-trust enable
# update-ca-trust
```

7. (Optional) If you are configuring IDM on an already existing Satellite Server or Capsule Server, the following steps should also be taken to make sure that the configuration changes take effect:

   a. Restart the foreman-proxy service:

```
# service foreman-proxy restart
```

   b. Log in to the Satellite Server and click **Infrastructure → Capsules**.

c. Click on the drop down menu on the right-hand side of the Capsule Server you have configured for IDM and choose **Refresh Features**.

8. Finally, create a new realm entry in the Satellite Server user interface:

a. Click **Infrastructure** → **Realms** and on the right-hand corner of the main page, click **New Realm**.

b. Fill in the fields in the following subtabs:

a. Realm - provide the realm name, the type of realm to use and the realm proxy.

b. Locations - choose the locations where the new realm is intended for use.

c. Organizations - choose the organizations where the new realm is intended for use.

c. Click **Submit**.

The Satellite Server or Capsule Server is now ready to provision hosts that automatically register to IDM. The next section will detail the steps on how to automatically add hosts to an IDM host group.

Report a bug

17.2. Adding Hosts to an IDM Host Group

Identity Management (IDM) supports the ability to set up automatic membership rules based on a system's attributes. Red Hat Satellite's realm feature provides administrators with the ability to map the Red Hat Satellite host groups to the IDM parameter "userclass" which allow administrators to configure automembership.

When nested host groups are used, they are sent to the IDM server as they are displayed in the Red Hat Satellite User Interface. For example, "Parent/Child/Child".

Note

The Satellite Server or Capsule Server sends updates to the IDM server, however automembership rules are only applied at initial registration.

1. On the IDM server, create a host group:

```
# ipa hostgroup-add hostgroup_name
Description: hostgroup_description
----------------------------
Added hostgroup "hostgroup_name"
----------------------------
  Host-group: hostgroup_name
  Description: hostgroup_description
```

Where:

a. *hostgroup_name* is the hostgroup's name.

b. *hostgroup_description* is the hostgroup's description.

2. Create an automembership rule:

```
# ipa automember-add --type=hostgroup automember_rule
----------------------------------
Added automember rule "automember_rule"
----------------------------------
Automember Rule: automember_rule
```

Where:

a. **automember-add** flags the group as an automember group.

b. **--type=hostgroup** identifies that the target group is a host group, not a user group.

c. *automember_rule* is the name you wish to identify the automember rule by.

3. Define an automembership condition based on the userclass attribute:

```
# ipa automember-add-condition --key=userclass --type=hostgroup --
inclusive-regex=^webserver hostgroup_name
----------------------------------
Added condition(s) to "hostgroup_name"
----------------------------------
  Automember Rule: automember_rule
  Inclusive Regex: userclass=^webserver
----------------------------
Number of conditions added 1
----------------------------
```

Where:

a. **automember-add-condition** allows you to add regular expression conditions to identify group members.

b. **--key=userclass** specifies the key attribute as userclass.

c. **--type=hostgroup** identifies that the target group is a host group, not a user group.

d. **--inclusive-regex=**^*webserver* is a regular expression pattern to identify matching values.

e. *hostgroup_name* is the target hostgroup's name.

When a system is added to the Satellite Server's *hostgroup_name* host group, it will now automatically be added to the Identity Management server's "*hostgroup_name*" host group as well. IDM host groups allow for Host-Based Access Controls (HBAC), sudo policies and other IDM functions.

Report a bug

Chapter 18. Red Hat Satellite User Interface Plugins

18.1. Red Hat Access Plugin

The Red Hat Access Plugin allows you to use a limited range of Red Hat Customer Portal services from the Red Hat Satellite web interface.

The following services are provided by the Red Hat Access plugin:

- **Search** provides the ability to search solutions in the Customer Portal from within the Red Hat Satellite interface.
- **Logs** provide users the ability to send specific parts of the log files to assist in problem solving. These log snippets can be sent to the Red Hat Customer Portal diagnostic tool chain.
- **Support** provides users the ability to access their open support cases, modify an open support case and open a new support case from within the Red Hat Satellite interface.

Note

To access Red Hat Customer Portal resources, you need to log in with your Red Hat Customer Portal user identification and password.

Report a bug

18.1.1. Searching for Solutions in the Red Hat Access Plugin

The Red Hat Access plugin provides search capabilities that will look through the solutions database available in the Red Hat Customer Portal without needing to log in to the Customer Portal interface.

To search for solutions from the Red Hat Satellite Server:

1. On the top right hand corner, click on **Red Hat Access** → **Search**.
2. Click **Log In** on the right corner of the main panel to log in to the Red Hat Customer Portal.

Note

To access Red Hat Customer Portal resources, you need to log in with your Red Hat Customer Portal user identification and password.

3. Enter your search query on the **`Red Hat Search:`** field. Search results display in the left-hand **`Recommendations`** list.
4. Click on a solution from the **`Recommendations`** list to view the solution article. The solution will display on the main panel.

Report a bug

18.1.2. Utilizing Logs in the Red Hat Access Plugin

The log file viewer gives Red Hat Satellite users visibility into the log files and allows users to isolate log snippets. These log snippets can then be sent through the Customer Portal diagnostic tool chain to help expedite problem solving.

To use the logs diagnostic tool from the Red Hat Satellite Server:

1. On the top right hand corner, click on **Red Hat Access → Logs**.
2. Click **Log In** on the right corner of the main panel to log in to the Red Hat Customer Portal. If you are already logged in, skip this step.

Note

To access Red Hat Customer Portal resources, you need to log in with your Red Hat Customer Portal user identification and password.

3. On the file tree on the left, select a log file and click the filename.
4. Click **Select File** to choose this file for viewing. A separate window will pop out with the log file contents.
5. Highlight any text sections of the log file. Highlighting text sections will activate the **Red Hat Diagnose** button.
6. Click **Red Hat Diagnose**. This will send the highlighted information to the Red Hat Customer Portal and provide solutions that closely match the provided log information.
7. If one of the solutions presented:
   - Match the problem you are experiencing, click the solution and follow the required steps to troubleshoot the issue.
   - Does not match the problem you are experiencing, click **Open a New Support Case**. The support case will be prepopulated with the highlighted text from the log file. See Section 18.1.5, “Creating New Support Cases Using the Red Hat Access Plugin”.

Report a bug

18.1.3. Viewing Existing Support Cases Using the Red Hat Access Plugin

To view existing support cases from the Red Hat Satellite Server:

1. On the top right hand corner, click on **Red Hat Access → Support → My Cases**.
2. Click **Log In** on the right corner of the main panel to log in to the Red Hat Customer Portal. If you are already logged in, skip this step.

Note

To access Red Hat Customer Portal resources, you need to log in with your Red Hat Customer Portal user identification and password.

3. Search for a specific support case from existing cases by:

a. Providing a key word or phrase in the **Search** field.

b. Choosing a specific **Case Group** from the drop down list. **Case Groups** are defined inside the Red Hat Customer Portal by your organization.

c. Choose a Case Status.

4. From the results, choose a specific support case and click on the **Case ID**.

Result:

The support case is ready for viewing.

Report a bug

18.1.4. Modifying Existing Support Cases Using the Red Hat Access Plugin

Prerequisites

Complete the instructions from the previous section.

Support Cases can be updated from the Red Hat Satellite Server web interface. When viewing the support case, scroll down to the sections marked as:

- **Attachments** - to attach a local file from the system. Add a filename to make it easier to identify.

Note

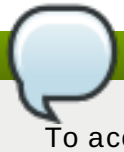
Filenames must be less than 80 characters. The maximum file size for web uploaded attachments is 250 MB. Use FTP for larger files.

- **Case Discussion** - to add any updated information about the case you wish to discuss with Global Support Services. Click on **Add Comment** after adding information into the comment field.

Report a bug

18.1.5. Creating New Support Cases Using the Red Hat Access Plugin

1. On the top right hand corner, click on **Red Hat Access** → **Support** → **New Case**.
2. Click **Log In** on the right corner of the main panel to log in to the Red Hat Customer Portal. If you are already logged in, skip this step.



Note

To access Red Hat Customer Portal resources, you need to log in with your Red Hat Customer Portal user identification and password.

3. The **Product** and **Product Version** fields are automatically populated. Fill in the other relevant fields:
   - **Summary** - a brief summary of the issue

- **Description** - a detailed description of the issue.

Note

Recommendations for possible solutions will appear on the main panel based on the summary typed in.

4. Click **Next**.
5. On the second screen, choose the appropriate:
   - **Severity** - Based on the urgency of the ticket, this can be classified as 4 (low), 3 (normal), 2 (high> and 1 (urgent).
   - Case Group - Based on who needs to be notified, case groups can be associated to the support case. Case Groups can only be selected in Red Hat Satellite. Case Groups are created within the Customer Portal.
6. Attach any required files. Add in a file description and click **Attach**.

   To make sure that relevant information is provided, it is recommended to attach the output of the following commands:

```
# sosreport
# foreman-debug
```

Important

foreman-debug removes all security information such as password, tokens and keys while collecting information. However, the tarball can still contain sensitive information about the Red Hat Satellite Server. It is recommended to send this information directly to the intended recipient and not publicly.

Note

Filenames must be less than 80 characters. The maximum file size for web uploaded attachments is 250 MB. Use FTP for larger files.

7. Click on **Submit**

Result:

The case is uploaded to the Customer Portal and a case number is provided for your reference.

Report a bug

Glossary of Terms

The following terms are used throughout this document. Familiarize yourself with these terms to help your understanding of Red Hat Satellite 6.

Activation Key

A registration token used in a Kickstart file to control actions at registration. These are similar to Activation Keys in Red Hat Satellite 5, but provide a subset of features because Puppet controls package and configuration management after registration.

Application Life Cycle Environment

An *Application Life Cycle Environment* represents a step, or stage, in a promotion path through the Software Development Life Cycle (SDLC). Promotion paths are also known as development paths. Content such as packages and Puppet modules move through life cycle environments by publishing and promoting Content Views. All Content Views have versions, which means you can promote a specific version through a typical promotion path; for example, from development to test to production. Channel cloning implements this concept in Red Hat Satellite 5.

Attach

The process of associating a Subscription to a Host that provides access to RPM content.

Capsule

A *Capsule* is an additional server that can be used in a Red Hat Satellite 6 deployment to facilitate content federation and distribution in addition to other localized services (Puppet Master, **DHCP**, **DNS**, **TFTP**, and more).

Compute Profile

Compute Profiles specify default attributes for new virtual machines on a compute resource.

Compute Resource

A *Compute Resource* is virtual or cloud infrastructure, which Red Hat Satellite 6 uses for deployment of hosts and systems. Examples include Red Hat Enterprise Virtualization Manager, OpenStack, EC2, and VMWare.

Content

Content includes software packages (RPM files) and Puppet modules. These are synchronized into the Library and then promoted into Life Cycle Environments using Content Views so that they can be consumed by Hosts.

Content Delivery Network (CDN)

The *Content Delivery Network (CDN)* is the mechanism used to deliver Red Hat content in a geographically co-located fashion. For example, content that is synchronized by a Satellite in Europe pulls content from a source in Europe.

Content View

A *Content View* is a definition of content that combines products, packages, and Puppet modules with capabilities for intelligent filtering and creating snapshots. Content Views are a refinement of the combination of channels and cloning from Red Hat Satellite 5.

External Node Classifier

An *External Node Classifier* is a Puppet construct that provides additional data for a Puppet Master to use when configuring Hosts. Red Hat Satellite 6 acts as an External Node Classifier to Puppet Masters in a Satellite deployment.

Facter

Facter is a program that provides information (facts) about the system on which it is run; for example, Facter can report total memory, operating system version, architecture, and more. Puppet modules enable specific configurations based on host data gathered by Facter.

Hammer

Hammer is a command line tool for Red Hat Satellite 6. Use Hammer to manage Red Hat Satellite 6 as a standard CLI, for scripts, and also through an interactive shell.

Hiera

Hiera is a key/value look-up tool for configuration data which allows keeping site-specific data out of puppet manifests.

Host

A *Host* refers to any system, either physical or virtual, that Red Hat Satellite 6 manages.

Host Collection

A *Host Collection* is equivalent to a Satellite 5 *System Group*, that is, a user defined group of one or more Hosts.

Host Group

A *Host Group* is a template for building a Host. This includes the content view (which defines the available RPM files and Puppet modules) and the Puppet classes to apply (which ultimately determines the software and configuration).

Location

A *Location* is collection of default settings that represent a physical place. These can be nested so that you can set up an hierarchical collection of locations. For example, you can set up defaults for "Middle East", which are refined by "Tel Aviv", which are further refined by "Data Center East", and then finally by "Rack 22".

Library

The *Library* contains *every* version, including the latest synchronized version, of the software that the user will ever deploy. For an Information Technology Infrastructure Library (ITIL) [1] organization or department, this is the Definitive Media Library [2] (previously named the Definitive Software Library).

Manifest

A *Manifest* transfers subscriptions from the Customer Portal to Red Hat Satellite 6. This is similar in function to certificates used with Red Hat Satellite 5.

For more information about certificates and subscription types, see:

- RHN Classic, Red Hat Satellite, and Channel Entitlements [3]
- The Structure of Satellite Certificates (Classic Style of Certificates) [4]

Organization

An *Organization* is an isolated collection of systems, content, and other functionality within a Satellite 6 deployment.

Permission

The ability to perform an action.

Product

A collection of content repositories. Products can be Red Hat products or newly-created products made up of software and configuration content.

Promote

The act of moving a content view comprised of software and configuration content from one Application Life Cycle Environment to another, such as moving from development to QA to production.

Provisioning Template

A *Provisioning Template* is a user-defined template for Kickstart files, snippets, and other provisioning actions. In Satellite 6 they provide similar functionality to Kickstart Profiles and cobbler Snippets in Red Hat Satellite 5.

Pulp Node

A *Pulp Node* is a Capsule Server component that mirrors content. This is similar to the Red Hat Satellite 5 Proxy. The main difference is that content can be staged on the Pulp Node before it is used by a Host.

Puppet Agent

The *Puppet Agent* is an agent that runs on a Host and applies configuration changes to that Host.

Puppet Master

A *Puppet Master* is a Capsule Server component that provides Puppet manifests to Hosts for execution by the Puppet Agent.

Repository

A *Repository* provides storage for a collection of content. For example, a YUM repository or a Puppet repository.

Role

A *Role* specifies a collection of permissions that are applied to a set of resources, such as Hosts.

Smart Proxy

A *Smart Proxy* is a Capsule Server component that can integrate with external services, such as **DNS** or **DHCP**.

Smart Variable

A *Smart Variable* is a configuration value that controls how a Puppet Class behaves. This can be set on a Host, a Host Group, an Organization, or a Location.

Standard Operating Environment (SOE)

A *Standard Operating Environment (SOE)* is a controlled version of the operating system on which applications are deployed.

Subscription

Subscriptions are the means by which you receive content and service from Red Hat.

Synchronizing

Synchronizing refers to mirroring content from external resources into the Red Hat Satellite 6 Library.

Synchronization Plans

Synchronization Plans provide scheduled execution of content synchronization.

User Group

A *User Group* is a collection of roles which can be assigned to a collection of users. This is similar to a Role in Red Hat Satellite 5.

User

A user is anyone registered to use Red Hat Satellite. Authentication and authorization is possible through built-in logic, through external LDAP resources, or with Kerberos.

Report a bug

---

[1] http://en.wikipedia.org/wiki/Information_Technology_Infrastructure_Library

[2] http://en.wikipedia.org/wiki/Definitive_Media_Library

[3] https://access.redhat.com/site/documentation/en-US/Red_Hat_Subscription_Management/1/html/MigratingRHN/sat-certs.html

[4] https://access.redhat.com/site/documentation/en-US/Red_Hat_Subscription_Management/1/html/Subscription_Concepts_and_Workflows/index.html#subscr-legacy

Revision History

Revision 1-18 **Fri Dec 19 2014** **David O'Brien**

Remove requirement for yum-rhn-plugin from chapter "Configuring Hosts".
Update some command layouts to comply with standards.

Revision 1-17 **Tues Dec 9 2014** **Megan Lewis**

BZ#1168273 Corrected the package name for Installing the Puppet Agent.

Revision 1-16.1 **Wed Nov 26 2014** **Athene Chan**

BZ#1139329 Added introductory text into "Using the Foreman Discovery Plugin".
BZ#1167966 Satellite Server backup script has changed, removed grinder from the command list.

Revision 1-16 **Mon Nov 24 2014** **Athene Chan**

BZ#1166660 Missing step in the configuring IDM chapter added.
BZ#1166656 Changed "realm-proxy@example.com" to "realm-capsule@example.com" for consistency.
BZ#1139329 Revised the Troubleshooting for the Foreman Discovery Plugin" section.

Revision 1-15.2 **Fri Nov 21 2014** **Athene Chan**

Removed the Foreman Discovery chapter for further review.

Revision 1-15 **Thurs Nov 20 2014** **Megan Lewis**

Minor corrections.
Added "Enabling Red Hat Repositories" section.

Revision 1-14 **Mon Nov 17 2014** **Megan Lewis**

Added further changes for BZ#1139329.

Revision 1-13 **Sun Nov 16 2014** **Megan Lewis**

BZ#1139329 Added chapter about Foreman Discovery.

Revision 1-12 **Fri Nov 14 2014** **Miroslav Svoboda**

BZ#1153596 Removed sentence mentioning support of Windows installation media.
BZ#1142477 Corrected procedure for Configuring Hosts for Registration.

Revision 1-11.2 **Friday Nov 14 2014** **Athene Chan**

BZ#1153567 Added a "Capsule Scalability" section.
BZ#1152797 Added a "Troubleshooting" section.

Revision 1-11.1 **Mon Nov 10 2014** **Athene Chan**

BZ#1150412 Added "--complete-reload" to the firewall-cmd firewall commands.
BZ#1141954 Changed "Installing the Katello Agent" to "Installing the Katello and Puppet Agents". Added information on puppet-agent in the section.

Revision 1-11 **Mon Nov 10 2014** **Athene Chan**

BZ#1161254 Added a new firewall rule to the list of firewall rules to allow katello-installer to run after initial install. Moved the firewall rules to the "Configuring Red Hat Satellite" sections to prevent error
BZ#1110837 Implemented QE edits.
BZ#1152630 Added RHEL7 firewall-cmd command examples for the firewall requirements.

Revision 1-10 **Fri Nov 7 2014** **Megan Lewis**

BZ#1149145 Defined the difference between All Hosts and Content Hosts and made sure all procedures pointed to the correct section.
Removed all instances of non breaking spaces in titles.

Revision 1-9 **Thu Nov 6 2014** **Athene Chan**

BZ#1110837 Added a "Configuring Identity Management" chapter in the User Guide.

Revision 1-8 **Thu Nov 6 2014** **Megan Lewis**

BZ#1149144 Corrected steps to locate systems registered via subscription-manager.

Revision 1-7 **Thu Oct 30 2014** **Megan Lewis**

Removed help file output.

Revision 1-6 **Thu Oct 23 2014** **Megan Lewis**

Implemented changes suggested by translation.

Revision 1-5 **Fri Oct 3 2014** **Athene Chan**

BZ#1140520 Changed all "ACME_Corporation" entries to the correct default organization entry "Default Organization".
BZ#1141954 Added example repositories to the "Enabling Red Hat Repositories" section and a note to enable RH Common repositories for client systems.
BZ#1140722 Added note to highlight that the command needs to change if the repository is different from the example command.

Revision 1-4 **Thu Oct 2 2014** **Megan Lewis**

Implemented brand changes.
Added Glossary of Terms in an Appendix.

Revision 1-3 **Wed Oct 1 2014** **Megan Lewis**

Minor edits based on feedback from translation.

Revision 1-2.01 **Fri Sep 12 2014** **Athene Chan**

BZ#1140875 Added firewall rules after the Satellite Server and Capsule Server installation.

Revision 1-2 **Fri Sep 12 2014** **David O'Brien**

Patched "Red Hat" entries to conform with Brand standards.

Revision 1-1 **Thu Sep 11 2014** **Athene Chan**

BZ#1140422 Changed the repository names for Red Hat Satellite Server and Red Hat Satellite Capsule Server.

Revision 1-0 **Tue 9 Sep 2014** **Megan Lewis**

Red Hat Satellite 6.0 GA Release.

Revision 0-23 **Thu 21 Aug 2014** **Megan Lewis**

BZ#1131654 - Removed optional from Step 4 in 15.2.1. Red Hat Satellite Backup Procedure.
BZ#1120722 - Corrected Step 2 in the note in 10.4.1. Registering a Host.
BZ#1131655 - Corrected database name in sections 15.2.1. Red Hat Satellite Backup Procedure an 15.2.2. Red Hat Satellite Restore Procedure.
BZ#1131613 - Section on creating a backup added to 1.3. Red Hat Satellite 6 Workflow.
BZ#1131604 - Section 15.2.1 - Removed "/var/lib/katello" from list for backup.

Revision 0-22 **Fri 15 Aug 2014** **Megan Lewis**

BZ#1120722 - Note in 10.4.1. Registering a Host corrected to reference a Host instead of a System.
BZ#1129841 - Added section 10.4.2. Installing the Katello Agent.
BZ#1127285 - Added prefix to baseurl used when registering clients to capsules.
BZ#1129578 - Removed sections 3.3.3 and 3.3.4.
BZ#1104431 - Implemented peer review feedback for Chapters 1-3.
Updated instructions for managing users and roles.
Updated instructions for using host collections.

Revision 0-21 **Tue 12 Aug 2014** **Athene Chan**

BZ#1128872 - Removed stray ; in Table 9.1.

Revision 0-20 **Fri 18 July 2014** **Athene Chan**

BZ#1120713 - Corrected section xml to prevent validation errors.

Revision 0-19 **Fri 11 July 2014** **Megan Lewis**

BZ#1109747 - Information added regarding organizations and subscription manifests.

Revision 0-18 **Thu 10 July 2014** **Athene Chan**

BZ# 1117861 - Section 10.3.1 Corrected CA Certificate URL.
BZ#1104914 - Section 5 Partial peer review implementation.

Revision 0-17 **Wed 9 July 2014** **Megan Lewis**

BZ#1116888 - Section 4.2.2.3 References to Katello CLI corrected to Hammer CLI.
BZ#1116543 - Section 10.3.1 Corrected RPM name.
BZ#1117503 - Section 5.3.1 Removed extra step.

Revision 0-16 **Wed 25 Jun 2014** **Athene Chan**

Preparing book for Beta release.

Revision 0-15 **Mon 11 Nov 2013** **Dan Macpherson**

Fixing minor issues.

Revision 0-14 **Mon 11 Nov 2013** **Dan Macpherson**

Preparation for MDP2.

Revision 0-13 **Wed 09 Oct 2013** **Dan Macpherson**

Adding table for synchronization content directories.

Revision 0-12 **Wed 09 Oct 2013** **Dan Macpherson**

Finalizing QE review implementation

Revision 0-11 **Tue 1 Oct 2013** **Athene Chan**

BZ#887680 Minor typo corrections.

Revision 0-10 **Mon 30 Sep 2013** **Dan Macpherson**

Rebuild from typo verification.

Revision 0-09 **Wed 18 Sep 2013** **Athene Chan**

Minor tagging errors corrected.

Revision 0-08 **Tue 17 Sep 2013** **Athene Chan**

BZ#956256, 969922, 864115 Implemented suggested changes to information on the User Guide.

Revision 0-07 **Fri 13 Sep 2013** **Athene Chan**

Book product changed.

Revision 0-06 **Thu 12 Sep 2013** **Athene Chan**

Minor grammatical edits.
Added book component to the ent file.

Revision 0-05 **Thu 12 Sep 2013** **Athene Chan**

BZ#1004566, 1004567, 1004568, 1004570, 1004571, 1004581, 1004586, 1004588, 1004590, 1004595, 1004597, 1004598, 1004600 Quality assurance edits implemented throughout the book.

Revision 0-04 **Mon 12 Aug 2013** **Dan Macpherson**

Removing draft watermark.

Revision 0-03 **Mon 12 Aug 2013** **Dan Macpherson**

Creating build for technical review.

Revision 0-02 **Tue 28 May 2013** **Athene Chan**

Initial book creation